

Modernization Techniques for IBM Power

Tim Simon	Bartlomiej Grabowski
Henry Vo	Mithun H R
Jordan Antonov	Hemraj Joshi
Marcelo Avalos Del Carpio	Niels Liisberg
Ian Bellinfantie	Arnold Ness
Andre Casagrande	Anil Patil
Carlo Castillo	Matthew Radford
Rafael Cezario	Sachin Sant
Paul Chapman	Mick Tarsel
Rohit Chauhan	Andy Youens



IBM Power

Artificial Intelligence



IBM Redbooks

Modernization Techniques for IBM Power

August 2025

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

First Edition (August 2025)

This edition applies to the following products:

- ▶ AIX 7.2 and 7.3
- ▶ IBM i 7.4, 7.5, and 7.6.
- ▶ IBM Power10-based processors
- ▶ Red Hat OpenShift 4.17 and 4.18
- ▶ PowerVM Virtual I/O Server (VIOS) 4.1.1.0

© Copyright International Business Machines Corporation 2025. All rights reserved.

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

Contents

Notices	ix
Trademarks	x
Preface	xi
Authors	xi
Now you can become a published author, too!	xvi
Comments welcome	xvi
Stay connected to IBM Redbooks	xvii
Part 1. Modernizing systems and applications in the IBM Power ecosystem	1
 Chapter 1. Defining modernization	3
1.1 Introducing modernization	4
1.2 What drives IBM clients to modernize	4
1.3 Benefits of modernization	5
1.3.1 Business benefits	6
1.3.2 Technical benefits	6
1.3.3 Cost optimization	7
1.3.4 Operational benefits	8
1.3.5 Data-driven decision making	9
1.4 Modernization in the IBM Power platform	10
1.4.1 Operating system support in IBM Power	13
1.4.2 Power Architecture benefits	20
1.4.3 Key benefits of IBM Power compared to x86 servers	23
1.5 Modernization components	24
1.5.1 Application modernization	24
1.5.2 Virtualization technologies	30
1.5.3 Containerization solutions on Power	41
1.5.4 Automation	44
1.6 IBM Power and artificial intelligence	47
1.6.1 Why artificial intelligence on IBM Power	47
1.6.2 Orchestrating AI flexibly	48
1.6.3 Safeguarding AI and data	48
1.6.4 Hybrid flexibility	48
1.6.5 Artificial intelligence solutions on IBM Power	49
 Chapter 2. Modernization considerations	51
2.1 Overview	52
2.2 Security	53
2.2.1 Encryption	55
2.2.2 Quantum encryption	56
2.3 Open-source software on IBM Power	57
2.4 Storage options	59
2.4.1 File storage	59
2.4.2 Block storage	60
2.4.3 Object storage	60
2.4.4 Container storage terminology and concepts	60
2.4.5 IBM Block Storage CSI driver	63
2.4.6 IBM Storage Scale	63

2.4.7 IBM Storage Fusion	66
2.4.8 IBM Fusion Deployment Options	69
2.5 High availability and disaster recovery	72
2.5.1 IBM solutions for HADR	73
2.5.2 IBM PowerHA SystemMirror	75
2.5.3 IBM Virtual Machine Recovery Manager	77
2.5.4 Linux high availability clustering	79
2.5.5 Additional information on HADR solutions	80
2.6 AI code assistants	80
2.6.1 IBM watsonx Code Assistant	80
2.6.2 IBM watsonx Code Assistant for Red Hat Ansible Lightspeed	83
2.6.3 Benefits of AI code assistants	83
Chapter 3. Client examples and use cases	87
3.1 IBM Power modernization references	88
3.2 Banking solutions	88
3.3 IBM CIO: Hybrid by Design	89
3.3.1 IBM CIO Cirrus	89
3.4 CI/CD in Cirrus	95
3.4.1 What CI/CD is	96
3.4.2 How CI/CD is implemented	96
3.4.3 A security example	96
3.5 Computer Systems Integration Ltd	98
3.5.1 IBM Power and x86 CIO Hybrid Cloud components	99
3.5.2 IBM Storage Scale and IBM Storage Fusion	101
Chapter 4. Services and consulting options	103
4.1 IBM Business Partners	104
4.1.1 IBM Partner Plus Directory	104
4.2 IBM Technology Expert Labs	106
4.3 Client Engineering	107
4.4 IBM Consulting	108
Chapter 5. Modernizing the management of IBM Power servers	111
5.1 HMC simplification	112
5.1.1 System and partition templates	112
5.1.2 Shared Ethernet Adapter simplification	112
5.1.3 Hardware Management Console update and upgrade	113
5.2 Virtual I/O Server management improvements	113
5.2.1 Updating VIOS	114
5.2.2 Upgrading VIOS	115
5.2.3 Automated Virtual I/O Server backups	115
5.2.4 Restoring VIOS	116
5.2.5 Validating the maintenance readiness of the Virtual I/O Server	116
5.2.6 Installing Virtual I/O Server images from the HMC	116
5.2.7 Microcode code update in I/O adapters	117
5.2.8 NFS-mounted ISOs in the Virtual Media Library	117
5.3 Power Virtualization Center overview	118
5.3.1 PowerVC architecture	120
5.3.2 PowerVC connectivity to managed systems	122
5.3.3 PowerVC automation	125
5.3.4 PowerVC setup tasks	126
5.3.5 VM deployment and automation scenarios	133
5.4 Ansible OpenStack Modules	135

5.4.1 An Ansible Execution Environment Image	136
5.4.2 Creating a collection	137
5.4.3 Building the collection	140
5.4.4 Uploading the collection to Ansible Hub	140
5.4.5 Create the custom Execution Environment Image file	143
5.4.6 Building the Ansible Execution Environment Image	148
5.4.7 Populating the environment variables	151
5.4.8 Using the EEI to run an Ansible playbook	152
5.5 Automation tools	156
5.5.1 Ansible	156
5.5.2 Terraform	159
5.5.3 Ansible Automation and Terraform	161
5.5.4 Terraform plan	162
5.5.5 Configuring and customizing Terraform defined resources	169
5.5.6 Ansible Terraform Collection	170
5.6 Power Enterprise Pools and Cloud Management Console	171
5.6.1 Power Enterprise Pools 1.0	172
5.6.2 Power Enterprise Pools 2.0	172
5.6.3 Cloud Management Console	172
5.7 PowerVM networking concepts	173
5.7.1 Shared Ethernet Adapter technology	174
5.7.2 Virtual network interface technology	175
5.7.3 SR-IOV and SEA comparison	176
5.7.4 vNIC configuration	179
5.7.5 vNIC and LPM	184
Chapter 6. How to modernize your applications	189
6.1 Modernizing systems through domain segmentation	190
6.2 Programming languages	193
6.2.1 Integrating traditional and modern technologies	194
6.2.2 Choosing a modern language for your project	195
6.2.3 Integrated development environments	196
6.3 Database technologies	198
6.3.1 Db2: The enterprise-grade relational backbone	198
6.3.2 Modern databases	198
6.3.3 Conclusion and comparison	199
6.4 CI/CD tools	201
6.4.1 GitHub	201
6.4.2 Red Hat OpenShift Pipelines	201
6.4.3 Red Hat OpenShift GitOps	202
6.5 Support for .NET on IBM Power	202
6.6 Kubernetes	203
6.6.1 Red Hat OpenShift	204
6.6.2 IBM Cloud Pak	204
Chapter 7. Tools and performance	207
7.1 Monitoring	208
7.1.1 Monitoring solutions	209
7.1.2 Using Prometheus and Grafana on Red Hat OpenShift with IBM Power	213
7.2 Logging	220
7.2.1 Traditional system logging	221
7.2.2 Container logging	221
7.2.3 Red Hat OpenShift Logging	222

7.2.4 Apache Log4j	223
7.3 Autoscaling and quality of service.....	225
7.3.1 Configuring quality of service for pods	225
7.4 Management.....	226
7.4.1 IBM Watson AIOps	226
7.4.2 Red Hat Advanced Cluster Manager	229
Part 2. Modernization support by operating system.....	237
Chapter 8. AIX	239
8.1 Introducing AIX	240
8.2 AIX advantages.....	240
8.2.1 Reliability, availability, and serviceability.....	241
8.2.2 Scalability and dynamic management.....	244
8.2.3 Advanced virtualization.....	245
8.3 AIX 7.3	245
8.3.1 Security updates.....	247
8.4 AIX processor compatibility modes	248
8.5 AIX and Power10 support	250
8.6 AIX support for open source	252
8.6.1 AIX Toolbox for Open Source Software	252
8.7 Live Update	263
8.7.1 Example use of Live Update.....	264
8.7.2 Live Update concepts	265
8.7.3 Performing a Live Update by using Network Installation Manager.....	268
Chapter 9. IBM i.....	279
9.1 IBM i modernization	280
9.1.1 Strategies and approaches to IBM i modernization	281
9.2 Modern development environments on IBM i	282
9.2.1 Visual Studio Code as a development tool for IBM i.....	282
9.2.2 Merlin	291
9.3 Db2 for i.....	301
9.3.1 Modernization techniques for IBM Db2 for i	301
9.3.2 Summary.....	306
9.3.3 References	306
9.4 Open-source databases on IBM i	306
9.4.1 MariaDB	308
9.4.2 PostgreSQL on IBM i	316
9.4.3 SQLite	320
9.4.4 Comparison.....	322
9.5 Traditional programming languages	323
9.5.1 RPG	323
9.5.2 COBOL.....	325
9.6 Open-source programming on IBM i.....	325
9.6.1 Available languages	325
9.6.2 Open-source programming support	326
9.7 Integrating traditional and open-source languages	327
9.7.1 Traditional languages to open-source languages.....	328
9.7.2 Open-source languages to traditional languages	333
Chapter 10. Linux and Red Hat OpenShift	343
10.1 Linux	344
10.1.1 Linux distributions that are supported on IBM Power	345

10.1.2 Linux on Power features and functions.....	347
10.2 Containerization solutions on IBM Power.....	351
10.2.1 Red Hat OpenShift	351
10.2.2 Kubernetes	353
10.2.3 Docker.....	356
10.2.4 Podman.....	358
Part 3. Appendixes	361
Appendix A. Components that are used in modernization on IBM Power	363
Red Hat	364
Red Hat OpenShift	364
Red Hat OpenShift Platform Plus	370
Middleware and application services	371
Run times	373
IBM.....	379
System management	379
IBM Cloud Pak	380
IBM watsonx	383
Open-source solutions	384
Kernel-based Virtual Machine.....	384
Artificial intelligence solutions	384
Pipelines	389
Databases.....	390
Frameworks	393
Security.....	395
Independent software vendors.....	396
Security.....	396
Finance	397
GigaSpaces Digital Integration Hub	399
Pipelines	399
MuleSoft Anypoint Flex Gateway	400
Backup and recovery	401
Observability	403
Dynatrace OneAgent and Operator	403
Splunk OpenTelemetry	404
Crest Infosolutions Alfresco	404
Appendix B. Modernization by using cloud-native tools	407
Overview	408
Skytap on Azure.....	408
What Skytap is	408
Azure native tools for modernization.....	411
Networking	411
Storage	413
DevOps for IBM i.....	414
Monitoring and analytics	421
Automation	422
AI with Azure	426
Appendix C. Details for performance claims	429
Improved performance	430
Running AI on a highly performant sustainable platform	430
Improved economics	430

Abbreviations and acronyms	433
Related publications	437
IBM Redbooks	437
Online resources	437
Help from IBM	438

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 <https://www.ibm.com/legal/copytrade.shtml>

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

AIX®	IBM Watson®	Redbooks®
Code Assistant™	IBM watsonx®	Redbooks (logo)  ®
Db2®	IBM Z®	Spyre™
DS8000®	Insight®	SystemMirror®
FlashCopy®	Instana®	Turbonomic®
GDPS®	Integrated Language Environment®	Turbonomic Application Resource Management™
Global Business Services®	Language Environment®	Watson Analytics®
Granite®	Micro-Partitioning®	watsonx®
HyperSwap®	Netcool®	watsonx Assistant™
IBM®	Orchestrate®	watsonx Code Assistant®
IBM API Connect®	Parallel Sysplex®	watsonx Orchestrate®
IBM Cloud®	POWER®	watsonx.ai®
IBM Cloud Pak®	Power Architecture®	watsonx.data®
IBM Consulting™	Power8®	watsonx.governance®
IBM FlashSystem®	Power9®	WebSphere®
IBM Instana™	PowerHA®	X-Force®
IBM Partner Plus®	PowerSC™	z/OS®
IBM Security®	PowerVC™	z/VM®
IBM Services®	PowerVM®	
IBM Spectrum®	pureScale®	
IBM Spyre™	Rational®	

The following terms are trademarks of other companies:

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

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.

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

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

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

VMware, VMware vSphere, 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

Recent global changes have impacted IT. Organizations are accelerating digital transformation to support an increasingly always-on world. IT leaders are focused on modernizing critical processes and operations to gain a competitive advantage in this dynamic environment.

Application modernization (updating applications to improve maintenance, extension, deployment, and management) is essential to meeting current and future needs. It provides numerous business and technical benefits.

Modernizing existing enterprise applications enables a smoother migration to hybrid cloud, offering the flexibility to run applications anywhere at any time. A cloud-native microservices approach maximizes the scalability and agility of the cloud.

Modernizing IBM Power allows new cloud-native microservices to integrate with existing applications by using the platform's performance, reliability, and security. This approach removes integration and productivity barriers, enabling enhanced user experiences, new applications, and new business opportunities.

This IBM Redbooks® publication provides a high-level overview of modernization, including key concepts and terminology to guide your modernization journey. It explores the components and architectural layers of the IBM Power ecosystem, demonstrating how they create an ideal platform for running mission-critical applications. The content is intended for business leaders, architects, and application developers.

Authors

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

Tim Simon is an IBM Redbooks Project Leader who is based in Tulsa, Oklahoma, US. He has more than 40 years of experience with IBM®, primarily in a technical sales role, helping customers design IBM solutions to address their business challenges. He holds a Bachelor of Science degree in Mathematics from Towson University in Maryland. Throughout his career, he has worked with a wide range of IBM products and has extensive experience developing customer solutions by using IBM Power, IBM Storage, and IBM Z®.

Henry Vo is an IBM Redbooks Project Leader with 11 years of experience at IBM. He has technical expertise in business problem solving, root cause analysis, and writing technical plans to support business objectives. Throughout his career at IBM, he has held multiple roles, including project management, ST/FT/ETE testing, back-end development, and serving as a DOL agent for New York. Henry is certified in IBM z/OS® Mainframe Practices, IBM Z System Programming, agile methodologies, and Telecommunication Development Jumpstart. He holds a master's degree in Management Information Systems (MIS) from the University of Texas at Dallas since 2012.

Jordan Antonov is based in Bulgaria and has worked at IBM for the past 10 years as part of the IBM Power support team. He is a member of the Product Engineering team. He holds a bachelor's degree in Machine Engineering from the Technical University of Sofia. His technical expertise includes IBM Power hardware, IBM PowerVM®, root-cause analysis, and hardware problem resolution. He has developed multiple technical documents and service education offerings.

Marcelo Avalos Del Carpio is a Open Group Architecture Framework (TOGAF)-certified Cloud Architect at Kyndryl Skytap Integration, supporting IBM accounts globally. Based in Uruguay, he has more than 10 years of IT experience, including roles at IBM. He specializes in the Power ecosystem, DevOps, cloud tools, modernization, and automation. He holds a degree in Electronic Systems Engineering from Escuela Militar de Ingeniería in Bolivia and a master's degree in Project Management from GSPM UCI in Costa Rica.

Ian Bellinfantie is the senior IBM AIX® and Red Hat Linux subject matter expert (SME) in the IBM Systems and Software Solutions Group at Saudi Business Machines (SBM), and the General Marketing and Services Representative of IBM World Trade Corporation. He has 29 years of experience provisioning IBM Power. Before joining SBM, he worked as an IBM contractual employee in the United Kingdom.

Throughout his career, he has gained extensive experience in deploying IBM software and hardware solutions, along with skills in programming and automation by using Shell, Perl, C++, Java, Chef, and Ansible. He holds a degree in Computer Science from the University of Greenwich and a master's degree in Financial Markets and Derivatives from London Metropolitan University.

His areas of expertise include PowerVM, IBM PowerVC™, IBM PowerHA®, Red Hat Enterprise Linux (RHEL), Red Hat Ansible, Red Hat OpenShift, AIX, and IBM Storage Scale. These skills have been primarily applied within the IBM Power installation base across the telecommunications, banking, retail, and public sectors. Ian brings this expertise and insight to the IBM Redbooks residency.

Andre Casagrande is a Modernization Specialist at IBM Brazil. He helps clients modernize their IBM Power infrastructure. With more than 20 years of experience in the tech industry, he focuses on Linux, AIX, and Free Software projects. His areas of expertise include DevOps, automation tools such as Ansible and Terraform, and digital transformation through the migration of monolithic applications to microservices.

Andre demonstrates products and integrations, including automation tools (Ansible and Terraform) and observability solutions such as IBM Instana®. He joined IBM in 2022, but his journey with the Power platform began in 2008 when he worked with an IBM Power 520 Express server, which sparked his interest in IBM technologies.

Since then, he has contributed to several product implementations, including Red Hat OpenShift on IBM Power, PowerVC, PowerVM, Virtual I/O Server (VIOS), and Hardware Management Console (HMC). He has also led infrastructure as code (IaC) projects, helping clients automate their infrastructure environments. Andre holds certifications as a Red Hat Certified Architect and a Site Reliability Engineer (SRE) through IBM, demonstrating his technical expertise and commitment to innovation.

Carlo Castillo is a Client Services Manager at Right Computer Systems (RCS), an IBM Business Partner and Red Hat partner that is based in the Philippines. He has more than 30 years of experience in pre-sales and post-sales support. His responsibilities include designing complete IBM infrastructure solutions, creating pre-sales configurations, performing IBM Power installation, implementation, and integration services, and delivering post-sales technical support. He also conducts presentations at customer engagements and corporate events.

In 1999, Carlo became the first IBM-certified AIX Technical Support Engineer in the Philippines. From 2007 to 2014, during RCS's tenure as an IBM Authorized Training Provider, he served as training coordinator. In this role, he administered the IBM Power curriculum and conducted training classes on AIX, PureSystems, PowerVM, and IBM i. Carlo holds a degree in Computer Data Processing Management from the Polytechnic University of the Philippines.

Rafael Cezario is a Senior Solutions Engineer at Blue Trust, an IBM Business Partner based in Brazil. He previously worked at IBM as a pre-sales technical specialist for IBM Power servers. With 19 years of experience in the IT industry, he has contributed to a wide range of infrastructure projects, including the design, implementation, demonstration, installation, and integration of solutions.

Rafael has worked with software on the IBM Power platform, including PowerVM implementations with Shared Ethernet Adapter (SEA) and virtual network interface cards (vNICs), PowerVC, IBM PowerSC™, Red Hat OpenShift, Ansible, and Network Installation Manager (NIM) servers. During his time at IBM, he served as a consultant for large clients, focusing on IBM Power and AIX. He also performed pre-sales and post-sales activities, and client presentations and demonstrations.

Throughout his career, Rafael has worked across multiple infrastructure domains and earned certifications in several technologies, including Cisco Certified Network Associate (CCNA), Nutanix Certified Associate (NCA), and IBM AIX. He holds a degree in Electrical Engineering with a specialization in Telecommunications from the Instituto de Ensino Superior de Brasília (IESB).

Paul Chapman is a Global Modernization Technical Leader for IBM Power Technology who is based in the United Kingdom. He has 28 years of technical and management experience working with IBM Business Partners and customers. Paul collaborates closely with Offering Management and Development Leaders to deliver first-of-a-kind projects and early adoption programs.

He led the launch of .NET on Power, created the Red Hat OpenShift Multi-Arch Compute Early Adoption Program, and contributed to the Development and Research Team's Co-Creation project. These efforts resulted in the first public references for .NET on Power and Red Hat OpenShift Multi-Arch Compute.

Paul regularly presents at conferences and shares technical knowledge through social media. He co-authored *Red Hat OpenShift V4.X and IBM Cloud Pak on IBM Power Systems Volume 2*, SG24-8486. He has received two Outstanding Technical Achievement Awards in recognition of his contributions.

Rohit Chauhan is an IBM Champion and Senior Technical Specialist at Tietoevry Tech Services in Stavanger, Norway. His areas of expertise are the IBM i platform and IBM Power. Rohit has more than 12 years of experience working with the IBM Power platform, focusing on the design, planning, and implementation of the IBM i infrastructure, which include delivering high availability and disaster recovery (HADR) solutions for a wide range of customers.

Before his current role, Rohit held technical leadership and security positions for IBM Power clients in Singapore and the United Arab Emirates. He has extensive experience in solution design, implementation, and system administration. He is an active member of Common Europe Norway, with a strong focus on the IBM i platform and security.

Rohit holds a bachelor's degree in Information Technology. He is an IBM Certified Technical Expert and holds an ITIL CDS certification. His areas of expertise include IBM i, Hardware Management Console (HMC), security enhancements, IBM PowerHA, system performance analysis and tuning, Backup Recovery and Media Services (BRMS), external storage, and PowerVM. He also provides IBM i platform solutions to clients and has co-authored several IBM Redbooks publications.

Bartłomiej Grabowski is an IBM Champion and Principal Systems Specialist at DHL IT Services, where he supports IBM Power globally. He has more than 20 years of experience in enterprise solutions. Bartłomiej holds a bachelor's degree in computer science from the Academy of Computer Science and Management in Bielsko-Biala, Poland. His areas of expertise include IBM Power, IBM i, IBM PowerHA, PowerVM, system performance, and storage solutions. He is a Platinum IBM Redbooks author.

Mithun H R is a Client Technical Architect at IBM Systems Development Labs in India. He has 14 years of experience supporting clients in modernization and digital transformation initiatives. His areas of expertise include performance optimization, benchmarking, complex integration, modernization, Red Hat OpenShift, and software development. Mithun has worked across various functional domains, including retail, aerospace, and defense. In his current role, he helps independent software vendors (ISVs) adopt cloud platforms and services. He also develops new workloads and benchmarks to demonstrate the competitive advantages of IBM platforms.

Hemraj Joshi is an IBM Enterprise Systems Presales Consultant at Gulf Business Machines (GBM) in Kuwait. He has more than 19 years of experience working with IBM technologies. Hemraj holds a master's degree in computer science from Pune University, India.

He specializes in IBM Power, IBM Storage Systems, IBM AIX, PowerVM, IBM PowerHA System Mirror, and storage area networks (SANs). He has extensive hands-on experience in the design, implementation, demonstration, and integration of enterprise solutions. His technical expertise spans a wide range of infrastructure projects, including the deployment of IBM and Brocade systems and IBM backup solutions.

Throughout his career, Hemraj has served as a subject matter expert (SME), contributing deep technical insights to numerous projects. He holds several IBM certifications and the Enterprise Architecture certification TOGAF. In his current role, Hemraj focuses on technical presales, working closely with clients to design tailored IBM solutions that address specific business challenges.

Niels Liisberg is the architect of the IceBreak application server for IBM i. He has developed IBM i middleware that is used in applications worldwide. He has contributed to the IBM i community through presentations, demonstrations, and open-source projects, including ILElastic, ILEVator, and noxDB. He focuses on moving traditional 5250 applications into modern architectures based on microservices and containers. He is a member of the Common Europe Advisory Council (CEAC) and has been recognized as an IBM Champion since 2019.

Arnold Ness is a Senior Power Technology Sales Leader in Canada. He has 40 years of experience at IBM and Ciena, where he has worked with clients to design and implement business solutions that use advanced technologies. Arnold holds a MBA in Information Technology Management from Royal Roads University, a degree in Electrical Engineering from the University of Alberta, and has completed the IBM Client Executive Program at Harvard University.

His areas of expertise include solution design and development across x86, IBM Power, and IBM Z platforms. In 2022, he received the Lou Gerstner Award for Client Excellence. He has implemented solutions for clients across North and South America. Arnold's professional interests include technology innovation, sustainability, hybrid cloud, artificial intelligence (AI), cybersecurity, Internet of Things (IoT), and quantum-safe computing.

Anil Patil is an Executive Architect and Solutioning Leader in Hybrid Cloud and Data at IBM Consulting™, US. He is a Certified Thought Leader in the Architect and Solution Consultant community, with 25 years of experience in design, development, architecture, and cloud migration for large and complex enterprise solutions. Anil's core expertise includes generative AI, Red Hat OpenShift, Amazon Web Services (AWS), and mainframe modernization. He is an IBM Redbooks author and a technical contributor to various IBM materials, external publications, and blogs. He holds a Bachelor of Engineering degree in Electronics and an Executive MBA in Finance and Strategy from Rutgers Business School in New Jersey.

Matthew Radford is a Remote Technical Support Specialist who is based in the United Kingdom. He works in IBM Power hardware support. He has been with IBM for 28 years in support roles. Matthew holds a Bachelor of Science (Honors) degree in Information Technology from the University of Glamorgan. His areas of expertise include 15 years of supporting IBM AIX and IBM PowerHA, and two years in IBM Power hardware support. He has co-authored several IBM Redbooks publications on IBM PowerHA and is recognized as an IBM Redbooks Gold Author.

Sachin Sant is a Linux on Power QA Architect at IBM India. He has more than 26 years of experience in operating system (OS) software development, including work on Linux, IBM AIX, and IBM OS/2. His background also includes server hardware validation for IBM Power and IBM Z systems, and open source-based test development and methodologies.

Sachin is an active contributor to several open source communities, including the Linux kernel and Linux test automation projects. In his current role, he is responsible for the functional quality of supported Linux releases on IBM Power servers. Previously, he led the Linux on IBM Z system software test team and was part of the Linux on Power Reliability, Availability, and Serviceability (RAS) software development team. He holds a degree in Electronics Engineering from Nagpur University, India.

Mick Tarsel is a Senior Linux Cloud Engineer with a strong focus on Linux networking. He has more than 10 years of experience working with virtualization software on IBM Power, with deep expertise in configuring, debugging, and optimizing network connectivity across diverse data center environments. Mick is an active contributor to open-source development. In 2019, he was published in the ICES Journal of Marine Science for his work using open-source software to support global coral reef conservation efforts. His professional interests include software-defined networking, operating system development, network security, and virtualization technologies.

Andy Youens is the Managing Director of FormaServe Systems, a UK-based IBM independent software vendor (ISV) since 1990. He is a seasoned IBM i professional with more than 40 years of experience in the field. Andy has deep expertise in IBM i systems, including system administration, application development, and modernization. Andy actively contributes to the community through speaking engagements, webinars, and technical articles. He is an IBM Champion and co-author of the IBM i Developer Certification.

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

Marc Bouzigues, Senior Solution Architect - Client Engineering - EMEA - IBM Power
IBM France

Jerome Calves, Senior Business Technology Leader - IBM Client Engineering | EMEA
IBM France

Ishwar Fernandes, IBM Champion, and Computer Systems Integration Ltd Head of Technical Architects
United Kingdom

Anandakumar Mohan, IBM Power, Senior Solution Architect, Technology Expert Labs
IBM India

Jenna Murillo, IBM Power ISV GTM and Technical Content Strategist
IBM Austin

Tim Rowe, STSM - IBM i Application Development & System Management
IBM Rochester

Alain Roy, Senior BTL - IBM Client Engineering EMEA
IBM France

Now you can become a published author, too!

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

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

Comments welcome

Your comments are important to us!

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

- ▶ Use the online **Contact us** review Redbooks form found at:
ibm.com/redbooks
- ▶ Send your comments in an email to:
redbooks@us.ibm.com

- ▶ Mail your comments to:
IBM Corporation, IBM Redbooks
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on LinkedIn:
<https://www.linkedin.com/groups/2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/subscribe>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<https://www.redbooks.ibm.com/rss.html>



Part 1

Modernizing systems and applications in the IBM Power ecosystem

IBM Power servers are renowned for their exceptional reliability and performance. They have long served as the backbone for critical business applications running on AIX, IBM i, and Linux. As organizations accelerate digital transformation, application modernization becomes essential to improve agility, reduce costs, and drive innovation.

IBM Power provides a robust platform for this transformation. It enables the modernization of earlier systems while preserving the platform's inherent reliability. With strong support for modern containerization technologies, including Red Hat OpenShift, and a powerful processor architecture that delivers higher workload capacity per core than competitors, IBM Power is suitable for modern IT environments.

This part explores how IBM Power supports clients in their modernization efforts. It highlights real-world examples, use cases, and tools that are available for managing modernized application stacks.

The following chapters are included in this part:

- ▶ Chapter 1, “Defining modernization” on page 3
- ▶ Chapter 2, “Modernization considerations” on page 51
- ▶ Chapter 3, “Client examples and use cases” on page 87
- ▶ Chapter 4, “Services and consulting options” on page 103
- ▶ Chapter 5, “Modernizing the management of IBM Power servers” on page 111
- ▶ Chapter 6, “How to modernize your applications” on page 189
- ▶ Chapter 7, “Tools and performance” on page 207



Defining modernization

Modernization is the process of upgrading and refining existing systems and applications to improve performance, efficiency, security, and overall business value. As businesses increasingly rely on technology to connect with customers, streamline operations, and remain competitive, meeting expectations for seamless digital experiences becomes critical. Earlier existing systems can hinder this progress, making modernization essential for technological upgrades, and transforming business operations and aligning IT infrastructure with evolving needs.

Because organizations generate large volumes of data, modern systems must process, analyze, and use this information effectively. Cloud platforms offer scalability, flexibility, and cost efficiency that traditional on-premises systems often cannot match. Also, the evolving security landscape demands stronger protection, which outdated systems frequently lack.

By modernizing applications and infrastructure, businesses can improve agility and foster innovation, enabling faster time-to-market for new products and services. Modernization also enhances the customer experience by enabling more personalized, responsive, and engaging interactions. It reduces costs through better resource usage and minimized downtime while also strengthening security to protect against cyberthreats.

With a well-planned modernization strategy, organizations can unlock greater value from their technology investments, thrive in the digital era, and gain a competitive advantage by using the latest technologies.

The following topics are described in this chapter:

- ▶ 1.1, “Introducing modernization” on page 4
- ▶ 1.2, “What drives IBM clients to modernize” on page 4
- ▶ 1.3, “Benefits of modernization” on page 5
- ▶ 1.4, “Modernization in the IBM Power platform” on page 10
- ▶ 1.5, “Modernization components” on page 24
- ▶ 1.6, “IBM Power and artificial intelligence” on page 47

1.1 Introducing modernization

In today's fast-paced digital environment, businesses face constant pressure to innovate, improve efficiency, and stay ahead of the competition. One of the most effective ways to meet these demands is through IT modernization, which is the process of upgrading, transforming, and optimizing technology infrastructure and applications. IT modernization goes beyond adopting new technologies; it involves rethinking business operations, aligning IT systems with evolving needs, and delivering seamless, secure, and scalable digital experiences.

Because organizations increasingly depend on technology to drive operations, manage data, engage customers, and deliver services, the limitations of earlier, existing systems become more evident. Earlier systems can hinder performance, compromise security, and create inefficiencies, making it difficult to respond quickly to market changes. IT modernization addresses these challenges by improving agility, reducing costs, enhancing security, and enabling better customer experiences.

Whether through cloud adoption, application updates, or the integration of emerging technologies, IT modernization is essential for organizations that aim to succeed in the digital age. By embracing modernization, businesses can unlock new opportunities, strengthen operational capabilities, and position themselves for long-term success.

1.2 What drives IBM clients to modernize

What drives IBM clients to modernize their IBM Power infrastructure? Figure 1-1 provides an overview of what clients are looking for in their modernized IT environment.

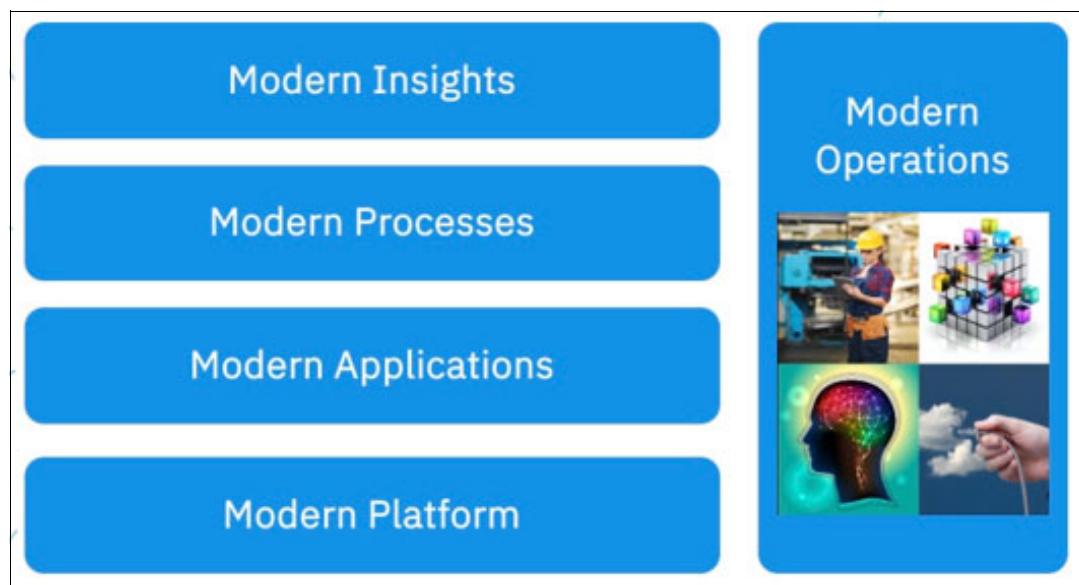


Figure 1-1 Drivers for modernization

Modern insights

In today's fast-paced market, IBM clients are modernizing to unlock the full potential of their data. They seek a unified data platform that enables seamless data sharing, real-time analytics, and deeper organizational insights. Enhanced access to data supports proactive decision-making, strengthens relationships with clients and suppliers, and helps organizations maintain a competitive edge.

Modern processes

In today's rapidly evolving landscape, IBM clients face increasing pressure to keep pace with change. To meet this challenge, they require automation that is powered by artificial intelligence (AI) and rule-based systems. These technologies help reduce response times, minimize decision-making risks, and enable organizations to adapt quickly to shifting market conditions.

Modern applications

IBM clients are moving away from cumbersome monolithic applications to improve user experiences and accelerate time-to-market. They are shifting toward distributed, application programming interface (API)-enabled architectures that support faster development cycles and simpler integration of new technologies. This approach results in more adaptable, scalable, and user-centric applications.

Modern platform

IBM clients are prioritizing infrastructure transformation to reduce total cost of ownership (TCO) and adopt flexible consumption models. They aim to minimize capital expenditures (CapEx), build a sustainable infrastructure footprint, and strengthen security in response to growing cyberthreats. In addition, clients are integrating advanced AI technologies to enhance application performance and streamline business processes.

Modern operations

To remain competitive in a rapidly changing world, clients must modernize their operations so that they can respond quickly to legislative and social changes, adapt to evolving user requirements, and reduce IT infrastructure costs. Modern application development is essential for achieving faster time-to-market, improved manageability, and reduced risk.

1.3 Benefits of modernization

Modernization enables organizations to improve competitiveness by increasing agility, reducing costs, and maximizing the value of existing IT investments. 83% of executives agree that modernizing applications and data is central to their organization's strategy.¹

Cloud migration strategies vary in effectiveness. Lift-and-shift migrations often fall short of cloud adoption goals.² A structured approach helps organizations determine whether rehosting, replatforming, or refactoring is the most effective path. Applications that use distributed computing patterns and service-oriented architectures typically require modifications, which improve the speed and cost efficiency of cloud migrations.

Modernization also contributes to cost reduction. Modernized infrastructures reduce unplanned outages, improving availability and mitigating reputational risk. Addressing architectural challenges, such as dynamic provisioning and latency, further enhances performance.

Scalability, resilience, and security are core outcomes of successful modernization. Cloud-native strategies accelerate application delivery, reduce time-to-market, and increase business agility. These improvements enable organizations to respond more quickly to customer demands and regulatory changes.

¹ Source:

<https://www.ibm.com/thought-leadership/institute-business-value/report/application-modernization-hybrid-cloud>

² Source: <https://www.gartner.com/en/documents/5585259>

The following sections explore the benefits of modernization from business, technical, operational, and data-driven perspectives.

1.3.1 Business benefits

Modernization supports business growth by enabling organizations to adapt quickly to market changes and meet evolving customer expectations.

Figure 1-2 illustrates the primary business benefits of modernization.

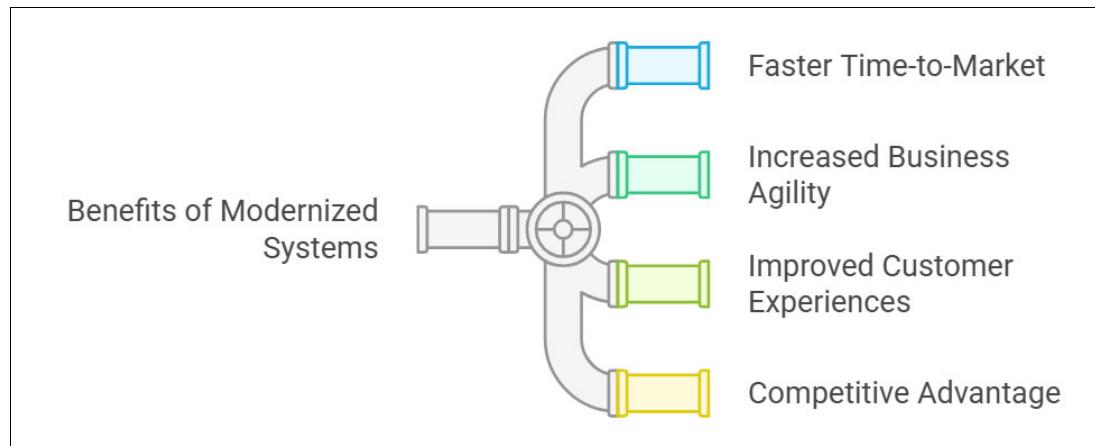


Figure 1-2 Benefits of modernized systems

- ▶ Faster time-to-market
 - Enables rapid development and deployment through continuous integration and continuous delivery (CI/CD) pipelines and automation. Organizations can meet market demands quickly while maintaining high quality.
- ▶ Increased business agility
 - Supports adaptability to changing needs, enabling organizations to scale, pivot strategies, and integrate new technologies. This flexibility helps address market shifts, customer demands, and regulatory changes.
- ▶ Improved customer experiences
 - Enhances application reliability, performance, and usability. Faster response times, intuitive interfaces, and personalized services lead to higher customer satisfaction and loyalty.
- ▶ Competitive advantage
 - Facilitates rapid innovation and industry differentiation. By integrating modern technologies such as AI, hybrid cloud solutions, and advanced analytics, organizations can achieve market leadership.

1.3.2 Technical benefits

Modernization enhances performance, improves interoperability, and helps ensure resilience. These benefits enable organizations to integrate new technologies and optimize their existing infrastructure.

Figure 1-3 on page 7 illustrates the primary technical advantages of modernization.

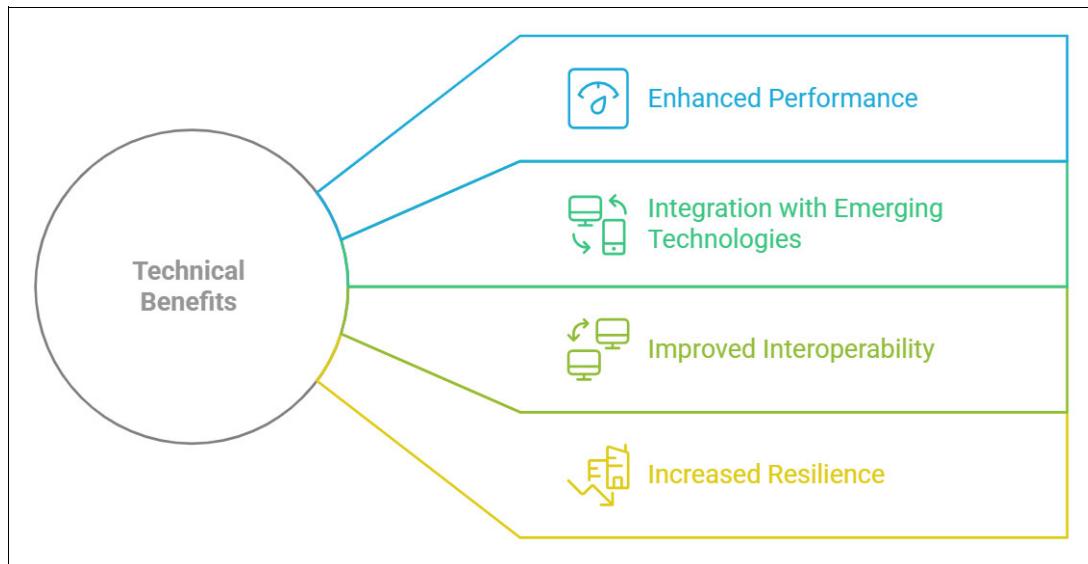


Figure 1-3 Technical benefits of modernized systems

- ▶ Enhanced performance

A modern architecture maximizes resource usage. By adopting a modern framework, organizations achieve faster processing speeds, improved reliability, and better scalability for applications and workloads.

- ▶ Integration with emerging technologies

Modern environments support integration with advanced technologies such as AI, machine learning (ML), and hybrid cloud services. This integration enables organizations to use cutting-edge tools to improve operations and drive innovation.

- ▶ Improved interoperability

Modern systems are designed to operate across diverse platforms, applications, and services. Improved interoperability helps ensure effective communication between systems and tools, simplifying processes and reducing operational silos.

- ▶ Increased resilience

Today's infrastructure is built to support disaster recovery (DR), high availability (HA), and fault tolerance. These capabilities help ensure continuous operations during failures or disruptions.

1.3.3 Cost optimization

Modernization delivers cost optimization benefits by reducing operational expenses while improving efficiency and sustainability.

Figure 1-4 highlights the primary cost-related advantages of modernized systems.

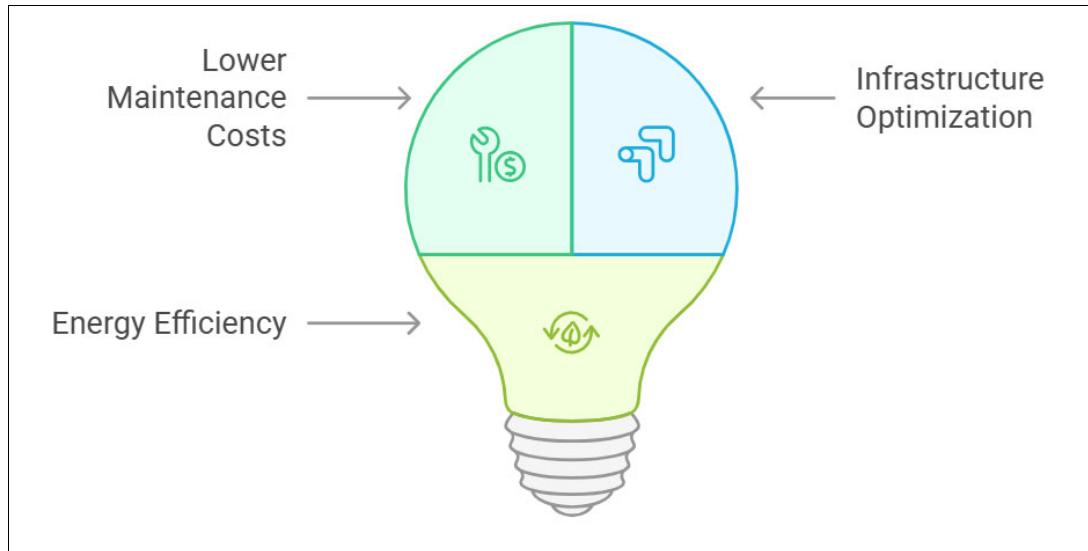


Figure 1-4 Cost optimization benefits of modernized systems

- ▶ Lower maintenance costs

By reducing reliance on aging hardware and software, which are often expensive to maintain, a modern infrastructure minimizes maintenance expenses. This migration also optimizes CapEx, helping enable organizations to allocate more resources toward strategic initiatives.

- ▶ Infrastructure optimization

Recent advancements in technology improve the usage of computing, storage, and networking resources. Virtualization and containerization enable scalable and flexible infrastructure, reducing TCO.

- ▶ Energy efficiency

A modern infrastructure uses less energy while delivering higher performance. Upgrading to energy-efficient hardware and software reduces utility costs and supports sustainability goals.

1.3.4 Operational benefits

Modernization enhances operations, security, and management, resulting in improved performance and greater control.

Figure 1-5 on page 9 illustrates the key operational benefits of modernization.



Figure 1-5 Operational benefits of modernized systems

- ▶ **Centralized management**

Modernization enables IT teams to manage multiple environments from a single control plane. Tools such as IBM Cloud Pak® and automated monitoring simplify administration, reduce management costs, and increase system visibility.

- ▶ **Workflow optimization**

Automation and streamlined processes enhance workflow efficiency. Tools such as Ansible for configuration management and Terraform for infrastructure as code (IaC) help automate tasks, reduce errors, and improve service delivery timelines.

- ▶ **Improved security**

Advanced encryption, multi-factor authentication (MFA), and continuous monitoring enhance. These capabilities provide robust security frameworks that comply with industry standards, protecting data and systems from emerging threats. Examples include the usage of AES-256 encryption and MFA protocols.

1.3.5 Data-driven decision making

Modernization equips organizations with tools that provide actionable insights and facilitate seamless data integration across platforms.

Figure 1-6 illustrates two key advantages: maximizing the value of data and enhancing data portability.

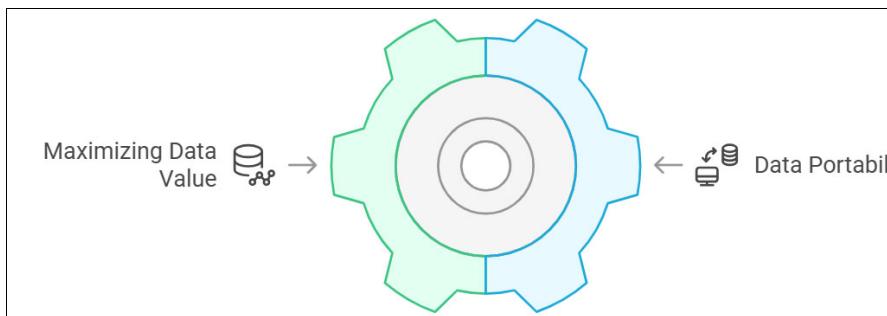


Figure 1-6 Data-driven benefits of modernized systems

- ▶ Maximizing data value

Modernized environments enhance the utility of enterprise data through advanced analytics and AI. These capabilities enable organizations to extract insights from large datasets, make data-driven decisions, optimize operations, and identify new business opportunities. For example, IBM Watson® Analytics® can forecast market trends or detect inefficiencies by using predictive analytics.

- ▶ Data portability

Modernized systems support seamless data transfer and integration across diverse platforms and environments. Data portability is essential for hybrid and multi-cloud strategies, allowing organizations to migrate workloads between on-premises systems and cloud platforms. It also supports compliance by enabling data access and reporting across regions and systems.

1.4 Modernization in the IBM Power platform

The IBM Power platform is a family of systems that can run mission-critical workloads by using hybrid multicloud technologies. IBM Power servers are high-performance, secure, and reliable systems built on the IBM POWER® processor architecture.

Figure 1-7 illustrates the Power10 family, starting with the one- and two-socket scale-out servers, scaling up to the four-socket IBM Power E1050, and culminating in the eight-socket IBM Power E1080.

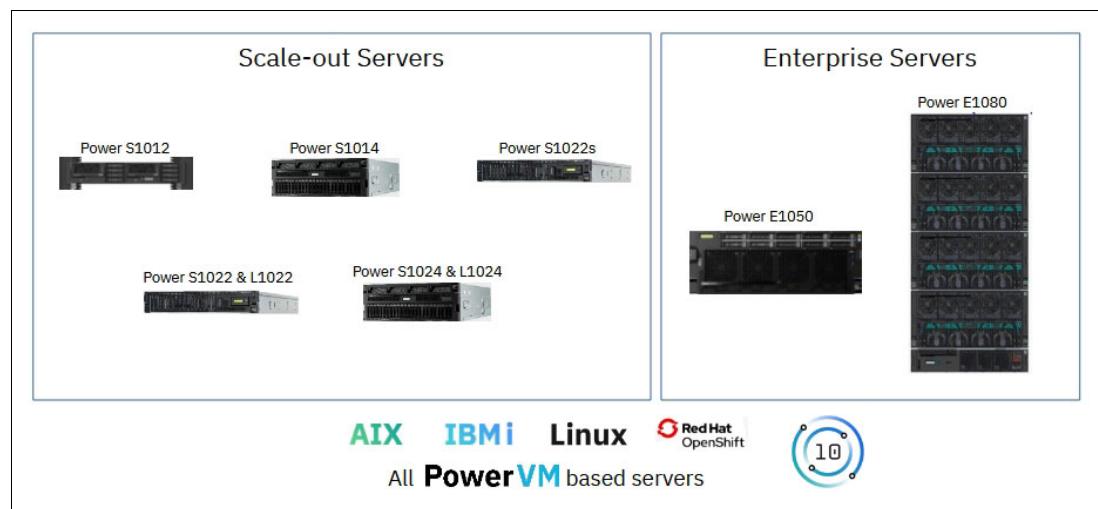


Figure 1-7 The Power10 family

The IBM POWER processor roadmap reflects a strong commitment to continuous innovation and performance enhancement. The Power10 processor, announced in August 2020 and released in September 2021, marked a significant leap in compute performance and energy efficiency. Power10 servers, such as the ones that are shown in Figure 1-7, are designed to handle mission-critical workloads with improved core-to-cloud data protection and streamlined automation. These systems deliver up to 75% performance improvements for the same workloads compared to previous generations.³

³ Source: <https://www.ibm.com/power>

Key features of Power10 include multiple packaging options: the single-chip module (SCM) that is used in the Power E1080, the dual-chip module (DCM) that is used in midrange and scale-out models, and the entry single-chip module (eSCM) that is used in low-end scale-out models. Other capabilities include PCIe Gen5 connectivity, transparent memory encryption, and support for advanced AI workloads through the Matrix Math Accelerator (MMA) feature, which accelerates matrix multiplication operations.

Looking ahead, the IBM Power11 processor (scheduled for release in 2025) promises further advancements. It integrates IBM Spyre™ Accelerator, which is designed for efficient AI computation and boosts AI processing capabilities. Power11 builds on the reliability, availability, and serviceability (RAS) features of Power10 while introducing improved energy efficiency and quantum-safe security. These enhancements position Power11 systems to support emerging enterprise AI use cases and drive digital transformation for mission-critical infrastructure.

Figure 1-8 illustrates recent generations of IBM Power processor-based systems and highlights IBM's continued investment in future innovation.

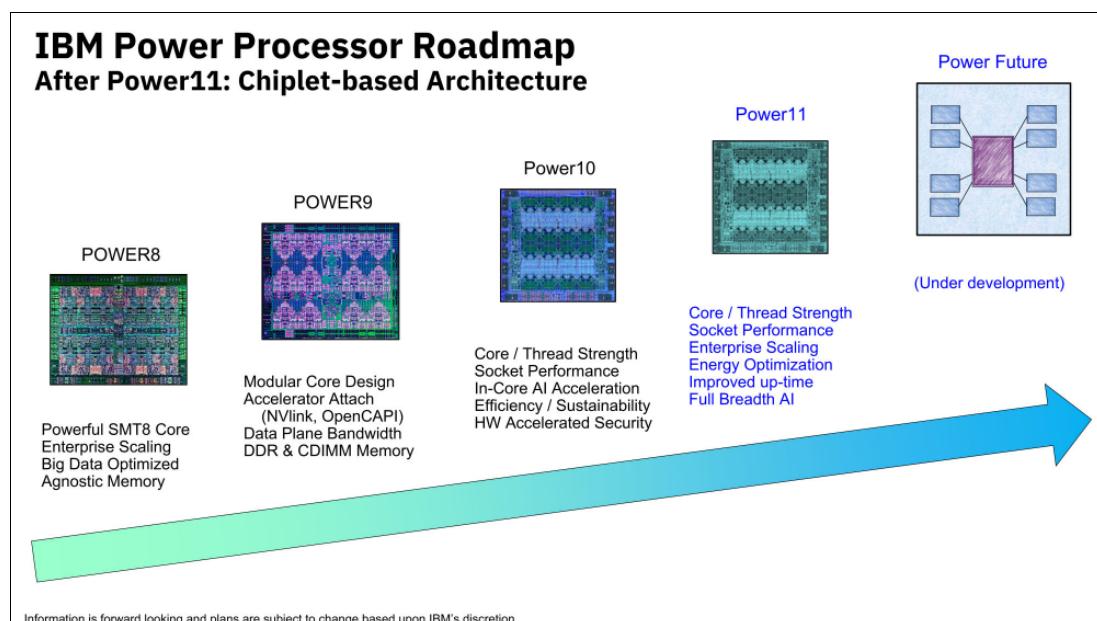


Figure 1-8 IBM Power family history and roadmap

Many of the world's most mission-critical enterprise workloads run on the IBM Power platform. This platform forms the backbone of a global IT infrastructure across industries such as finance, retail, government, and healthcare. IBM Power servers are recognized for their industry-leading security, reliability, and performance. Clients rely on the IBM Power platform for enterprise applications that include AI, enterprise resource planning (ERP), databases, and application and web servers.

Digital transformation is reshaping enterprise IT delivery, with cloud computing playing a central role. Organizations require flexibility in how they use infrastructure, and the IBM Power platform is designed to support both on-premises and cloud deployments. Whether modernizing enterprise applications with Red Hat OpenShift and Kubernetes, building a private cloud with pay-as-you-go services, deploying applications on IBM Cloud®, or managing a hybrid multicloud environment, the IBM Power platform provides the capabilities to support any hybrid cloud strategy.

The modern data center is a blend of on-premises and off-premises environments, incorporating multiple platforms such as the IBM Power platform, IBM Z, LinuxONE, and x86. Applications range from monolithic to cloud-native, running on a combination of bare metal, virtual machines (VMs), and containers. An effective hybrid cloud management solution must accommodate this diversity. IBM and Red Hat are positioned to support both current and modernized applications, regardless of where they are.

Figure 1-9 provides a view of the IBM Power E1080, which is the high-end enterprise model in the IBM Power product portfolio.



Figure 1-9 View of the Power E1080 central processor component drawer

The IBM Power platform delivers one of the highest availability ratings among servers

According to the ITIC 2023 Global Server Hardware Server OS Reliability Survey,⁴ which polled nearly 1,900 corporations across more than 30 vertical market segments, 88% of IBM Power10 users reported achieving eight nines (99.999999%) of uptime. This percentage equates to only 315 milliseconds of unplanned outage time per server per year due to system flaws or component failures, second only to IBM Z, which reported 31.56 milliseconds of annual downtime per server. As a result, Power10 enterprise users spend only \$7.18 per server per year on remediation for unplanned outages that are caused by hardware or component flaws. 2023 marks the 15th consecutive year that IBM Z and LinuxONE, and the IBM Power platform led the industry in uptime reliability across 18 mainstream server platforms.

This exceptional availability is largely attributed to the built-in recovery and self-healing capabilities of Power10. These features reduce downtime by enabling redundancy and fault tolerance. Also, organizations can migrate from earlier Power servers to Power10 servers without interrupting application availability, helping ensure HA during transitions.

⁴ Source: <https://itic-corp.com/itic-2023-reliability-survey-ibm-z-results/>

IBM Power is consistently rated as one of the most secure servers in the market

For the fourth consecutive year, IBM Power was rated as one of the most secure systems in 2022,⁵ with unplanned outages due to security issues that total 2.7 minutes or less annually. This level of security makes IBM Power:

- ▶ 2x more secure than HPE Superdome servers
- ▶ 6x more secure than Cisco UCS servers
- ▶ 16x more secure than Dell PowerEdge servers
- ▶ 20x more secure than Oracle x86 servers
- ▶ Over 60x more secure than unbranded white box servers

In addition, 95% of surveyed IBM Power servers detected security breaches immediately or within the first 10 minutes. This rapid detection reduces the risk of downtime, data compromise, or loss, helping organizations maintain operational continuity and data integrity.

IBM Power enables businesses to boost operational efficiency to meet sustainability goals

A recent user case study demonstrated how IBM Power enabled a client to improve user application performance by 20% while also achieving their sustainability goals. By migrating to IBM Power and IBM FlashSystem® storage, the client was able to fully use their SAP S/4HANA operations. This modernization effort supported both operational efficiency and climate objectives.

IBM Power streamlines AI operations with advanced on-chip technologies

IBM Power servers deliver up to 5x faster AI inferencing per socket for high-precision math compared to the previous generation. This performance gain is enabled by multiple Matrix Math Accelerator (MMA) units that are embedded in each IBM POWER processor core. These MMAs eliminate the need for external accelerators (such as GPUs and their associated device management) when running ML and inferencing workloads.

Current IBM Power servers offer a wide range of scalability. Entry-level scale-out servers, such as the IBM Power S1012, start with 1 core and 32 GB of memory, while high-end enterprise systems, such as the IBM Power E1080, scale up to 240 cores and 64 TB of memory.

Note: The full lineup of IBM server models that are based on the latest IBM POWER processors is available at [IBM Power](#).

1.4.1 Operating system support in IBM Power

IBM Power offers unparalleled flexibility by allowing you to consolidate diverse operating environments onto a single system. From industry-leading options such as AIX and IBM i to widely adopted platforms like Linux and Red Hat OpenShift, you can use IBM Power to consolidate mission-critical applications across multiple systems.

⁵ Source:

<https://techchannel.com/backup-and-recovery/ibm-z16-and-power10-deliver-highest-reliability-among-mainstream-servers-for-15th-consecutive-year/#:~:text=The%20IBM%20z16%20and%20Power10,saves%20money%20and%20mitigates%20risk>

This approach provides the following benefits:

- ▶ Enhanced reliability, availability, and security
- ▶ Faster response to business demands
- ▶ End-to-end data protection, from core to cloud
- ▶ Streamlined insights and automation

You can modernize your applications and infrastructure with a seamless hybrid cloud experience. IBM Power servers deliver the agility, reliability, and sustainability your organization needs.

Supported operating systems

At the time of writing, Power10 processor-based systems support the platforms and operating system (OS) versions that are listed in Table 1-1.

Table 1-1 Power10 operating system support matrix

Operating system	Supported versions
Red Hat OpenShift Container Platform	4.9 or later
PowerVM Virtual I/O Server (VIOS)	4.1.0.0 or later 3.1.2.30 or later 3.1.1.50 or later
AIX	7.3 TL0 or later (with any I/O configuration) 7.2 TL4 or later (with any I/O configuration) 7.1 Technology Level 5 (TL5) or later (through VIOS only)
IBM i	7.6 7.5 7.4 TR5 or later 7.3 TR11 or later
Red Hat Enterprise Linux (RHEL)	8.4 or later 9.0 or later
SUSE Linux Enterprise Server	15.3 or later 12.5
Ubuntu	22.04 or later

Note: The reference system that is used in Table 1-1 is the IBM Power E1080. Software maps that detail which versions are supported on specific IBM Power server models, including previous generations, are available on the [IBM Support page](#).

A complete list of supported operating systems is also available [here](#).

AIX

IBM AIX is IBM's proprietary OS that is designed to run on IBM Power servers. The first POWER processors (then part of the RISC System/6000 (RS/6000) platform) ran AIX 3. Earlier versions of AIX were available on previous IBM hardware, such as the RT/PC. At the time of writing, the available versions of AIX on IBM Power are AIX 7.2 and AIX 7.3.

For more than three decades, organizations have relied on IBM AIX to run their most mission-critical applications. AIX on Power supports innovation through hybrid cloud and open-source capabilities, enabling you to build and deploy modern applications in a secure and resilient environment.

Today, nearly every organization recognizes that digital transformation is essential to improving customer service, reducing costs, enhancing operational efficiency, and advancing environmental sustainability. To support these goals, businesses are modernizing both infrastructure and applications by developing or migrating workloads to the cloud and adopting technologies such as containerization. The most successful organizations understand that this transformation requires a hybrid cloud approach.

To accelerate this journey, over the past 2 years IBM introduced new offerings, including IBM Power Hybrid Cloud capabilities with on-premises IBM Power Private Cloud and consumption-based pricing, and IBM Power Virtual Server. IBM also expanded its support for open-source technologies by bringing popular tools to AIX and developing a collection of Ansible automation packages and playbooks. These tools simplify AIX management by using consistent skills and processes across x86-based platforms.

For more than 35 years, AIX and IBM Power have supported mission-critical workloads and databases for tens of thousands of customers. The platform continues to lead the industry in performance, scalability, resiliency, flexibility, and security. IBM remains committed to evolving AIX to help customers take advantage of emerging capabilities, such as running Red Hat OpenShift containers alongside AIX to reduce latency, or embedding AI inference into enterprise applications on AIX.

AIX release roadmap

AIX on Power supports innovation through hybrid cloud and open-source capabilities, enabling you to build and deploy modern applications in a secure and resilient environment. AIX remains a strategic and foundational component of the IBM Power portfolio, with a roadmap and support plan that extends beyond 2035.

Figure 1-10 illustrates this roadmap.

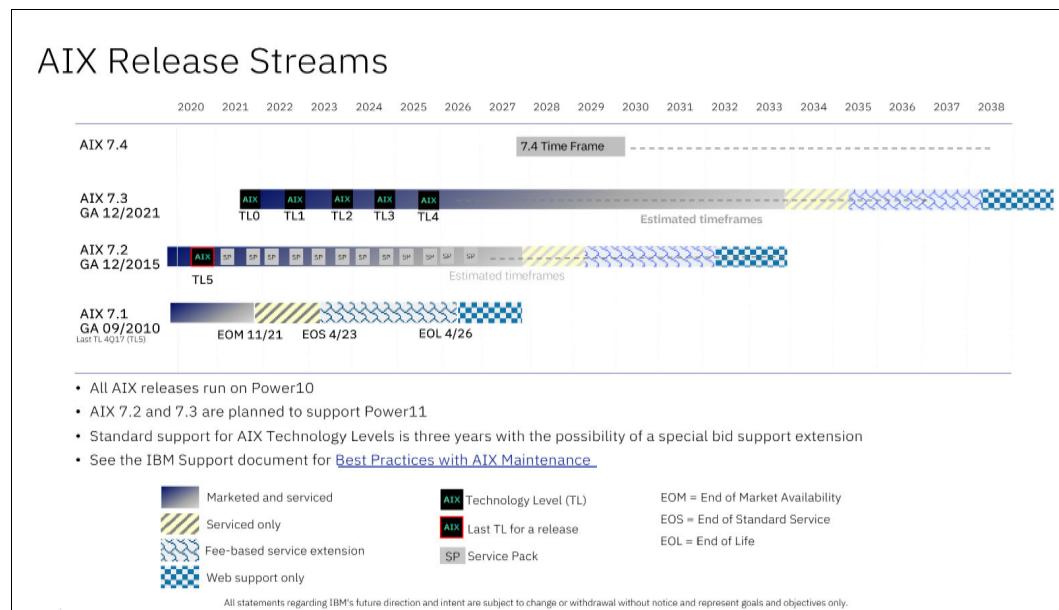


Figure 1-10 AIX release roadmap

Benefits of using AIX

A few of the notable benefits of using AIX are as follows:

- ▶ Security leadership: AIX delivers robust, long-term security with features such as Trusted AIX and Trusted Execution (TE).
- ▶ Unmatched uptime: Power servers that run AIX 7.3 offer industry-leading reliability and the lowest percentage of unplanned annual server downtime.
- ▶ Investment protection: AIX helps ensure binary compatibility, allowing applications to run unchanged and without recompilation on the latest releases.
- ▶ Enterprise AI: AIX 7.3 enables AI inference to run directly in the core, streamlining insights and accelerating decision-making.

AIX binary compatibility

AIX binary compatibility allows applications that are created on earlier releases or technology levels of AIX to run unchanged and without recompilation on later versions. For example, an application that is developed on AIX 6.1 can run on AIX 7.2 or later.

This capability, which is known as compatibility with an earlier version, requires applications to use only portable programming techniques to help ensure binary compatibility across platforms.

For more information about AIX binary compatibility, see [IBM Documentation](#).

A system that runs AIX 7.3 or later can serve as a server for client machines running earlier versions of AIX. In such cases, the server must have the necessary compatibility options installed. All binary compatibility conditions apply in this scenario.

Note: If applications do not run correctly after they migrate to a later version of AIX, you can open a case with IBM Support. Be sure to include *AIX Binary Compatibility* in the Title field when you submit the case.

Restrictions that are related to AIX binary compatibility are also outlined in [IBM Documentation](#).

For more information about AIX, see [IBM AIX](#).

IBM i

IBM i, formerly known as AS/400 and iSeries, offers a range of benefits that make it a strong choice for businesses, especially ones that run mission-critical applications. Its lineage traces back to IBM's earliest midrange systems for small to medium-sized businesses, and its capabilities expanded over time. Today, some of the world's largest organizations use IBM i on IBM Power servers as a strategic platform across industries such as manufacturing, retail, distribution, logistics, banking, healthcare, insurance, hospitality, government, and legal services.

IBM i is a powerful and versatile platform that provides a secure, scalable, and efficient foundation for running applications and managing data. Its emphasis on integrated solutions, openness, and built-in value makes it an attractive option for organizations that seek a robust and reliable operating system and database platform.

As a fully integrated operating system, IBM i includes the database, middleware, security, runtime, and hypervisor, which are all licensed as a single stack. This integration helps reduce TCO, simplifies system management, and enables organizations to do more with fewer resources.

According to Forrester Consulting, clients that deploy IBM i either on-premises or in the cloud achieved an average return on investment (ROI) of 191% and a payback period of 6 months. These clients also reported savings of \$1.06 million from reduced system downtime and a productivity gain of \$470,000 over 3 years.⁶

IBM i roadmap

A unique feature of IBM i is its compatibility with earlier versions. Organizations can run AS/400 applications on IBM i without encountering compatibility issues. This capability eliminates the need for costly and time-consuming code migrations.

IBM i also benefits from a strong ecosystem of users, developers, and independent solution providers, offering a wide range of tools and services to support business needs.

Figure 1-11 illustrates the supported versions of IBM i at the time of writing and the roadmap for future releases.

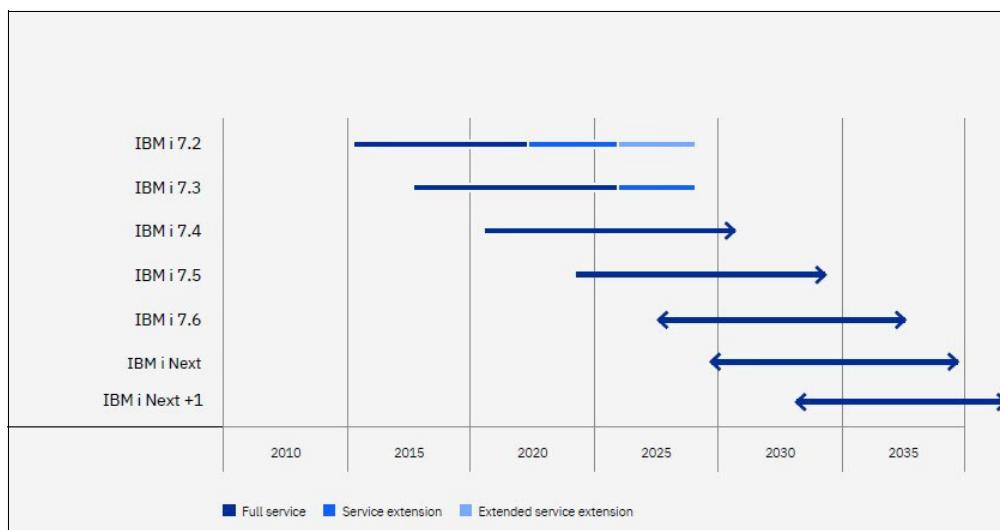


Figure 1-11 IBM i release schedule

Subscription licensing

Subscription-term licensing provides clients with predictable and consistent payment options, making annual budget planning simpler. Software licenses and support are bundled into a single subscription price. Annual subscriptions are automatically updated to the latest release, helping reduce business risk and security vulnerabilities by keeping systems current. With auto-renewal, clients no longer need to manage software keys or worry about lapses in support. Clients on earlier releases can return to support and modernize their environments at a lower cost.

In summary, IBM i is a robust and versatile operating system that offers a unique combination of reliability, security, cost-efficiency, and modernization capabilities. It is a strong choice for businesses that seek a stable and secure platform for mission-critical applications while also enabling them to adopt new technologies and adapt to evolving business needs.

For more information about IBM i, see the [IBM i product page](#).

⁶ Source: <https://www.ibm.com/account/reg/us-en/signup?formid=urx-52179Forrest>

Linux on Power

With IBM systems and Linux open-source software, you can select the optimal hardware for your workloads while managing software costs effectively. Linux is a highly regarded operating system that offers flexibility, stability, and a low TCO. As an open-source development project, Linux benefits from continuous innovation that is driven by a global community.

IBM supports Linux as a long-term strategic platform and actively contributes to its development through the Linux Technology Center (LTC).

Linux is certified on all IBM systems, including IBM Power, IBM Z, and IBM LinuxONE, allowing you to choose the hardware that best fits your business needs.

Table 1-2 lists the Linux distributions that are supported on IBM Power10 servers.

Table 1-2 Linux distributions for Power10 processor-based systems

IBM Power10 Systems	Distributions supported
9043-MRX (IBM Power E1050) 9105-22A (IBM Power S1022) 9105-22B (IBM Power S1022s) 9105-41B (IBM Power S1014) 9105-42A (IBM Power S1024) 9786-22H (IBM Power L1022) 9786-42H (IBM Power L1024)	RHEL 9.0 and later RHEL 9.x releases RHEL 8.4 and later RHEL 8.x releases SUSE Linux Enterprise Server 15 SP3 and later SUSE Linux Enterprise Server 15 updates Red Hat OpenShift Container Platform 4.9 or later Ubuntu 22.04 or later ^a
9080-HEX (IBM Power E1080)	RHEL 9.0 and later RHEL 9.x releases RHEL 8.4 and later RHEL 8.x releases RHEL 8.2 (IBM Power9® Compatibility mode only) ^b SUSE Linux Enterprise Server 15 SP3 and later SUSE Linux Enterprise Server 15 updates SUSE Linux Enterprise Server 12 SP5 (Power9 Compatibility mode only) Red Hat OpenShift Container Platform 4.9 or later Ubuntu 22.04 or later ^a
9028-21B (IBM Power S1012)	RHEL 9.2, for PowerLE or later Red Hat OpenShift Container Platform 4.15 or later SUSE Linux Enterprise Server 15 SP6 and later SUSE Linux Enterprise Server 15 updates Ubuntu 22.04 or later ^a
IBM Power10 processor-based systems support the following configurations per logical partition (LPAR):	
<ul style="list-style-type: none">▶ SUSE Linux Enterprise Server 15 SP4: Up to 64 TB of memory and 240 processor cores.▶ SUSE Linux Enterprise Server 15 SP3: Up to 32 TB of memory and 240 processor cores.▶ RHEL 8.6, or later: Up to 64 TB of memory and 240 processor cores.▶ RHEL 8.4 and 9.0: Up to 32 TB of memory and 240 processor cores.▶ SUSE Linux Enterprise Server 12 SP5 and RHEL 8.2: Up to 8 TB of memory and 120 processor cores.	

a. Ubuntu on Power support is available directly from Canonical.

b. Red Hat Business Unit approval is required for using RHEL 8.2 on IBM Power10 processor-based systems.

Note: For more information about Linux distributions that are supported on Power, see this [IBM Documentation article](#).

Python packages in Linux on IBM Power

You can lower the entry barrier for AI by natively using Python packages on Linux running on IBM Power LPARs. No container platform is required, enabling seamless integration with existing IBM Db2®, SAP, and Oracle environments. You can use more than 200 Python packages that are optimized for IBM Power10.

Note: For more information, see [RocketCE in a Linux on Power LPAR](#).

SAP RISE

The RISE with SAP methodology provides SAP customers with a clear and confident path to SAP Business Suite.⁷

In February 2023, SAP and Red Hat, one of the world's leading providers of open-source solutions, announced an expanded partnership to increase SAP usage of and support for RHEL. This collaboration aims to enhance intelligent business operations, support cloud transformation across industries, and drive holistic IT innovation.⁸ SUSE Linux Enterprise Server remains supported too.

RHEL is a hardened, production-ready Linux operating system that is designed for hybrid cloud innovation. Trusted by global enterprises, RHEL provides a consistent and reliable foundation for SAP software deployments. It serves as a standard Linux backbone for SAP clients that operate in hybrid and multi-cloud environments. SAP is also migrating an increasing portion of its internal IT infrastructure and SAP Enterprise Cloud Services portfolio on RHEL.

IBM Power servers are purpose-built for data-intensive applications such as SAP HANA and SAP S/4HANA. These workloads require high-performance in-memory computing while maintaining the HA and flexibility that are needed for hybrid cloud environments.

With global data volumes projected to exceed 180 zettabytes by 2025, organizations across all sectors face growing pressure to manage, process, store, and extract insights from critical data. Running SAP HANA on IBM Power servers helps reduce data center costs and improve environmental sustainability:

- ▶ The IBM Power E1050 uses 50% less energy than comparable x86-based servers with similar performance.
- ▶ The IBM Power E1080 uses 15% less energy and delivers 54% more performance at maximum input power compared to equivalent x86-based servers.

For more information about SAP HANA running on IBM Power, see [SAP HANA on IBM Power](#).

Red Hat OpenShift Container Platform

Red Hat OpenShift is the industry's leading enterprise-grade Kubernetes platform, capable of running in any environment, such as on-premises in your data center, on IBM Cloud, or on third-party cloud providers such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud. It supports multiple architectures, including ppc64le, s390x, x86_64, and AMD platforms.

⁷ Source: <https://www.sap.com/mena/products/erp/rise.html>

⁸ Source:

<https://www.redhat.com/en/about/press-releases/sap-and-red-hat-deepen-partnership-power-sap-software-workloads-red-hat-enterprise-linux#:~:text=SAP%20and%20Red%20Hat%20announce,RISE%20with%20SAP%20solution%20deployments>

Red Hat OpenShift is optimized to enhance developer productivity and foster innovation. It is fully supported on all IBM Power servers starting with IBM Power9 processors when running RHEL CoreOS 4.14 or later.

For greater flexibility, Red Hat OpenShift can be paired with Red Hat OpenShift Data Foundation or IBM Storage Fusion to deliver a software-defined storage (SDS) solution. This combination simplifies cloud transformation projects.

1.4.2 Power Architecture benefits

Application modernization comes in many shapes and sizes, and you might not know where to start. This section describes how the IBM Power platform brings strengths and benefits to your modernization efforts. There are many more benefits than the ones that are enumerated here. IBM Power is built for core enterprise applications and the next wave of digital transformation that is fueled by application modernization. Here are a few advantages of modernizing with IBM Power10.

Application modernization takes many forms, and identifying a starting point can be challenging. This section explains how IBM Power contributes strengths and benefits to your modernization initiatives. The advantages that are listed here are not exhaustive. IBM Power is designed for core enterprise applications and supports the next wave of digital transformation that is driven by application modernization. Here are several key benefits of modernizing with IBM Power10:

- ▶ Pervasive security and resiliency
- ▶ More performance from software with fewer servers
- ▶ Superior performance for your enterprise data
- ▶ Flexible and efficient usage
- ▶ Incremental application modernization
- ▶ Innovating with an extensive container software ecosystem

Pervasive security and resiliency

To address today's security challenges, every layer of your organization's IT hardware and software stack must remain secure. IBM Power provides the most reliable mainstream server platform, enabling innovation and faster time to market without compromising security.

The IBM Power multi-layered security approach offers full visibility into your hardware and software. With IBM Power10 hardware-accelerated transparent memory encryption, quantum-safe cryptography, and Fully Homomorphic Encryption (FHE), your data is protected by comprehensive, end-to-end security across every layer of the stack that safeguards against both current and future threats.

More performance from software with fewer servers

You can run an equivalent set of applications at comparable throughput levels while using fewer IBM Power servers than competing platforms. IBM Power delivers a 55% lower 3-year TCO for modern cloud-native applications, achieving 4.4x better per-core throughput.⁹ This efficiency enables the colocational of cloud-native applications with existing AIX, IBM i, and Linux VM-based workloads, providing low-latency API access to business-critical data.

Also, you can reduce containerized software licensing costs, such as for IBM Cloud Pak, by using subcapacity licensing through PowerVM shared processor pools (SPPs). This approach

⁹ Source:

<https://www.ibm.com/products/blog/10-reasons-why-ibm-power10-is-the-trusted-foundation-for-modernization#:~:text=You%20can%20buy%20fewer%20IBM,connections%20to%20business%2Dcritical%20data>

allows CPU cores to be autonomously shared across Red Hat OpenShift worker nodes without compromising application performance.

Superior performance for your enterprise data

Running Red Hat OpenShift in a VM alongside your AIX, IBM i, or Linux VMs enables low-latency, reliable communication with enterprise data through PowerVM VIOS. This configuration improves performance by reducing network hops. It also enhances security by keeping network traffic within the physical server, allowing secure communication between cloud-native applications and enterprise data stores.

Flexible and efficient usage

You can manage demand spikes and support more cloud workloads per server by using IBM PowerVM Hypervisor (PHYP) with on-demand CPU and memory capacity. IBM Power virtualization technology dynamically allocates resources by sharing pools of CPU cores across nodes. Differentiating hypervisor and consumption features, such as uncapped processors and SPPs, enable performance service-level agreements (SLAs) while reallocating unused processor cycles to worker nodes that require more capacity.

On-premises pay-as-you-go consumption is available for Red Hat OpenShift running on IBM Power.

Incremental application modernization

With IBM Power10, teams can modernize existing AIX, IBM i, and Linux applications incrementally by extending them with new cloud-native services in a secure and methodical way. This approach enables you to capitalize on existing investments in applications and skills while driving incremental transformation, which reduces costs, accelerates time-to-value, and minimizes risk.

For IBM i clients, this process is further simplified with the IBM i Modernization Engine for Lifecycle Integration (Merlin). Merlin is a set of tools that run in Red Hat OpenShift containers and guide developers through the modernization of IBM i applications.

Innovating with an extensive container software ecosystem

A strong software ecosystem is essential to any application modernization effort, enabling teams to innovate with the latest technologies. Open-source communities now play a critical role in supporting organizational modernization initiatives.

IBM Power supports core business applications and a wide range of open-source and commercial container software that run on Red Hat OpenShift. Choosing IBM Power for modernization means gaining industry-leading reliability, performance, and security, along with superior compute capabilities for data-intensive and mission-critical workloads. It provides a robust foundation for modern, container-based applications.

Trusted foundation

Kubernetes provides the foundation for modernizing enterprise applications. As the industry's leading enterprise Kubernetes platform, Red Hat OpenShift offers a consistent environment for application development and containerized workloads across hybrid cloud, multicloud, and edge deployments, which benefits both developers and IT administrators.

Developers gain access to the latest software innovations within Red Hat OpenShift to accelerate solution development, and IT administrators can efficiently observe, operate, and manage the platform and infrastructure. This approach enables the faster delivery of high-value, high-quality software to users.

Red Hat OpenShift Container Platform

Red Hat OpenShift is the industry's leading enterprise-ready Kubernetes platform, capable of running anywhere, such as on-premises in your data center, on IBM Cloud, or on third-party cloud providers such as AWS, Azure, or Google Cloud.

Optimized to enhance developer productivity and foster innovation, Red Hat OpenShift is fully supported on all IBM Power servers starting with IBM Power9 processors when running RHEL CoreOS 4.14 or later.

For greater flexibility, Red Hat OpenShift can be paired with Red Hat OpenShift Data Foundation or IBM Storage Fusion to deliver an SDS solution that simplifies cloud transformation projects.

IBM Cloud Pak and Red Hat software

IBM Power delivers superior performance and cost efficiency for containerized workloads, including IBM Cloud Pak and a broad set of Red Hat open-source software solutions. These tools support both the modernization of existing applications and the development of new cloud-native applications on Red Hat OpenShift.

IBM Cloud Pak offers three key benefits:

- ▶ Comprehensive and simple to use
- ▶ Supported by IBM
- ▶ Deployable anywhere Red Hat OpenShift runs

IBM Cloud Pak offerings accelerate modernization by bundling everything that is needed to get started. Available IBM Cloud Pak offerings on IBM Power include the following ones:

- ▶ IBM Cloud Pak for Applications
- ▶ IBM Cloud Pak for Data (CP4D)
- ▶ IBM Cloud Pak for Watson AIOps (Infrastructure Automation)
- ▶ IBM Cloud Pak for Integration (CP4I)
- ▶ IBM Cloud Pak for Business Automation (CP4BA)

With multi-architecture clusters, you can now integrate more Cloud Pak capabilities into IBM Power processor-based Red Hat OpenShift clusters that use x86 worker nodes.

From a Red Hat software perspective, a comprehensive set of solutions is also available to accelerate modernization efforts, which include Red Hat Runtimes, Red Hat 3scale API Management, Red Hat Fuse, and Red Hat AMQ.

Figure 1-12 on page 23 illustrates the robust portfolio that is available for modernizing applications on IBM Power.

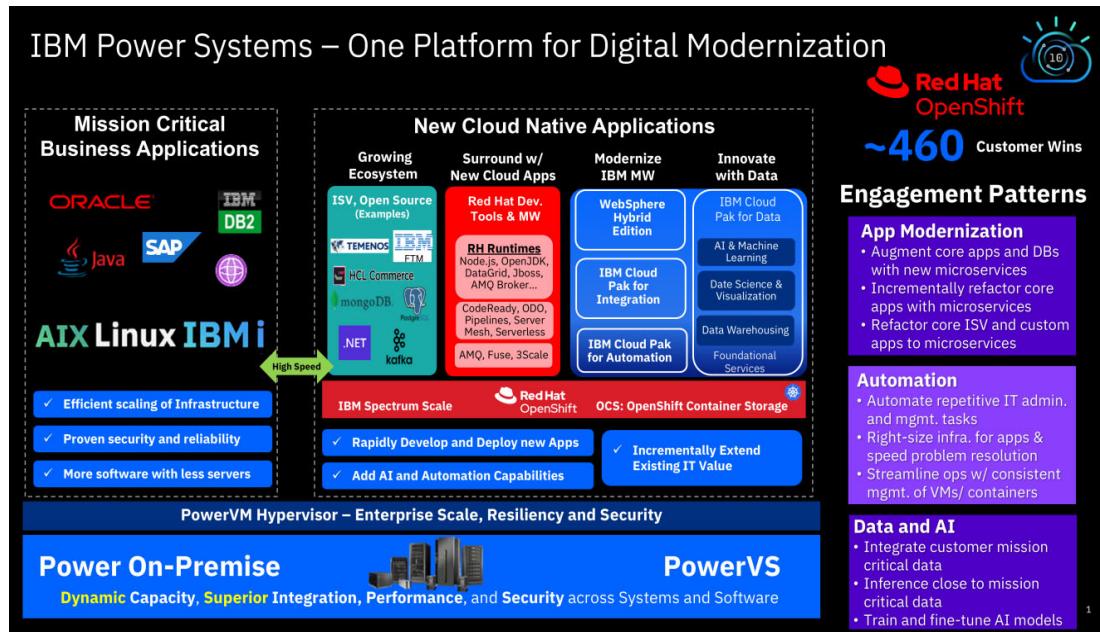


Figure 1-12 Modernization portfolio

Comprehensive hybrid cloud management and automation

As organizations increasingly adopt hybrid cloud IT models, the need for consistent management, observability, and automation becomes critical. Consistency across hardware platforms, cloud environments, and operating systems is essential for both IT administrators and developers.

IBM and Red Hat address these needs with a suite of integrated solutions, including IBM Cloud Pak for Watson AIOps (Infrastructure Automation), IBM Instana Observability, IBM Turbonomic Application Resource Management™ (ARM), Red Hat Advanced Cluster Management for Kubernetes, and Red Hat Ansible Automation Platform. These tools extend the value of the IBM Power platform, making it simpler to operate and automate hybrid cloud environments.

1.4.3 Key benefits of IBM Power compared to x86 servers

It is a common perception that x86 servers offer the best foundation for cloud computing due to their lower upfront acquisition costs. However, this view often overlooks critical factors such as performance, scalability, energy efficiency, data center space usage, reliability, and overall manageability. When these elements are considered, the TCO and ROI for x86 servers typically fall short compared to IBM Power servers.

IBM Power provides a more robust and cost-effective platform, offering superior value through its unique combination of performance and enterprise-grade features:

- ▶ SAP SD two-tier benchmark results show IBM Power with 8 sockets (120 cores) outperforming an x86 platform with 16 sockets (448 cores).¹⁰
- ▶ IBM Power delivers per-core performance that is 2.5 times faster than Intel Xeon Platinum.¹¹

¹⁰ Source: <https://www.sap.com/dmc/exp/2018-benchmark-directory/#/sd>

¹¹ Source: <https://www.spec.org/cpu2017/results/cpu2017>

- ▶ When running containerized applications and databases on IBM Power E1080, compared to x86 servers, IBM Power achieves the following results:
 - 48% lower 3-year TCO
 - 4.3 times more throughput per core
 - 4.1 times better price-performance
- You have fewer servers, which leads to a fourfold reduction in data center footprint, software licenses, and energy consumption.¹²
- ▶ In the ITIC 2023 Global Server Hardware and OS Reliability Survey, IBM Power was ranked the most reliable platform for the 15th consecutive year, outperforming all Intel x86 platforms and second only to IBM Z. IBM Power also reported only one data breach during the same period, compared to higher numbers on x86 platforms.¹³
- ▶ As shown in Table 1-1 on page 14, IBM Power supports a wide range of AIX, IBM i, and Linux workloads simultaneously, offering unmatched virtualization flexibility compared to x86 systems.

Although both IBM Power and x86 architectures are mature platforms for modern workloads, IBM Power stands out for its efficiency, deeply integrated virtualization, exceptional reliability, and unmatched scalability. These features enable support for enterprise-class workloads with less infrastructure than what is required by x86 hardware.

1.5 Modernization components

Modernization enhances IT infrastructure, applications, and workflows to improve agility, efficiency, and scalability. This section highlights key focus areas, including application modernization, containerization, and automation.

1.5.1 Application modernization

Application modernization involves transforming applications to use modern architectures, improving maintainability, scalability, and interoperability. This process might include rehosting, replatforming, refactoring, or replacing applications based on business and technical requirements.

Understanding application types is essential for defining an effective modernization strategy. Applications typically fall into the following categories:

- ▶ Traditional (monolithic) applications
- ▶ Cloud-native applications
- ▶ Composite applications

Each of these application types is described in the following sections.

Traditional (monolithic)

Monolithic applications consist of tightly coupled components within a single, self-contained architecture. This model traditionally served as the foundation of enterprise IT. However, it presents challenges in scalability, maintainability, and agility. A monolithic application is a single, indivisible unit in which all components are tightly integrated, typically running inside a VM, also known as an LPAR.

¹² Source: <https://www.ibm.com/it-infrastructure/resources/power-performance/e1080/#5>

¹³ Source: <https://itic-corp.com/itic-2023-reliability-survey-ibm-z-results/>

Figure 1-13 shows an example of a monolithic application.

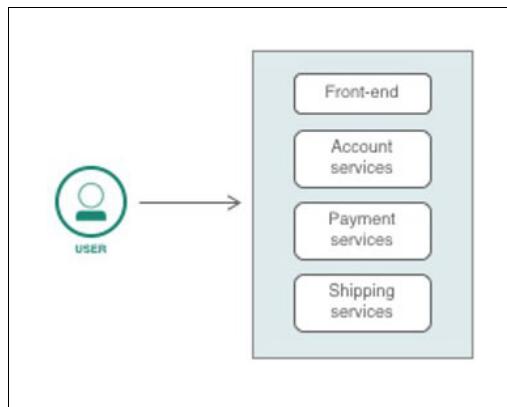


Figure 1-13 Monolithic application model

A monolithic application contains all the required layers for operation:

- ▶ User interface (UI): Manages what the user sees, including images, text, and other elements that are displayed through the UI.
- ▶ Business logic: Encodes the business rules that determine how data can be manipulated.
- ▶ Data access layer: Provides simplified access to data that is stored in persistent storage.
- ▶ Application integration layer: Handles integration with other services or data sources.

Monolithic applications feature a centralized architecture with interconnected components. They include complex dependencies between the business logic, UI, and data storage, which limits adaptability in cloud-based or distributed environments. These applications are also characterized by the usage of a waterfall development methodology with predefined release cycles.

Figure 1-14 illustrates these challenges and highlights the architectural barriers that traditional monolithic applications face when adapting to modern IT environments.

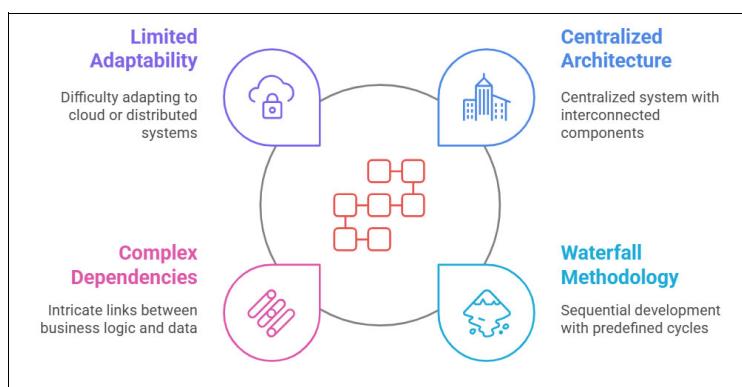


Figure 1-14 Challenges of monolithic applications

Key characteristics of monolithic applications

Here are the key characteristics of monolithic applications:

- ▶ Single codebase: All code that is required for the application is in one central location. This structure simplifies development by enabling communication in a single format and supporting work in a shared environment.
- ▶ Tightly combined: All components are interconnected. Changes to one part of the application can unintentionally affect other parts or the entire system.
- ▶ Self-contained: Traditional monolithic applications are designed to operate independently.

Advantages and disadvantages of monolithic applications

Here are the advantages of monolithic applications:

- ▶ Simpler development: With a single codebase, development is more straightforward, especially for smaller projects with well-defined requirements.
- ▶ Deployment simplicity: Monolithic applications are typically deployed as a single unit, which simplifies the integration process.
- ▶ Testing and debugging: A unified codebase simplifies testing and debugging for small to medium-sized applications due to reduced complexity and streamlined processes. However, in large and complex applications, these tasks can become time-consuming due to increased interdependencies.
- ▶ Improved security: The closed-system architecture limits access to internal processes and data, offering a higher level of protection against cyberthreats.

Here are some of the disadvantages of cloud-native applications:

- ▶ Limited agility: Tightly coupled components restrict the ability to introduce changes or implement new features without refactoring. Changes in one area can affect the entire application.
- ▶ Reduced scalability: Scalability is a major challenge. Even minor adjustments, such as modifying a single function, might require rebuilding the entire system, which is time-consuming and labor-intensive.
- ▶ Technology constraints: Monolithic applications are often limited to a single technology stack.

Cloud native

In a cloud environment, the demand for new functions, runtime integration, and continuous resource scaling, along with the need for applications to be deployable on new LPARs or containers, presents challenges that monolithic applications struggle to meet. This limitation stems from their inherent development structure, which complicates maintenance and updates.

Cloud-native applications are modular, scalable, and agile. They use containerization, microservices, and API-driven integration. These applications align with DevOps methodologies to support continuous deployment (CD) and rapid adaptation to evolving business needs.

Cloud-native applications represent the evolution from monolithic architectures to agile, scalable, and API-centric solutions. This migration follows structured modernization stages, including modular design, API integration, cloud deployment, and CI/CD automation. These elements enable organizations to adopt cloud-neutral platforms, enhance system interoperability, and support modern development frameworks.

Figure 1-15 shows the architecture and deployment of cloud-native applications.

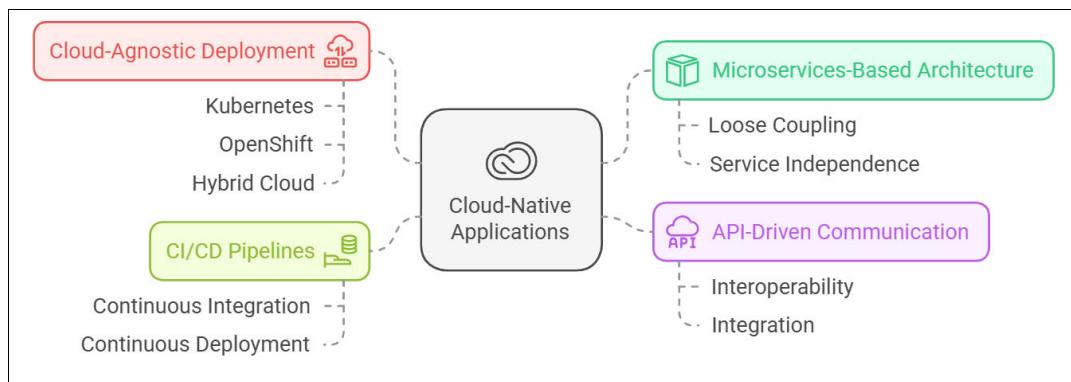


Figure 1-15 Cloud-native application architecture

Cloud-native architecture, also known as the microservices architecture, is an approach in which a single application is built by using smaller, independently deployable components instead of a single monolithic block. These smaller components are referred to as *services*. Each function of the application is deployed as a separate service, as shown in Figure 1-16.

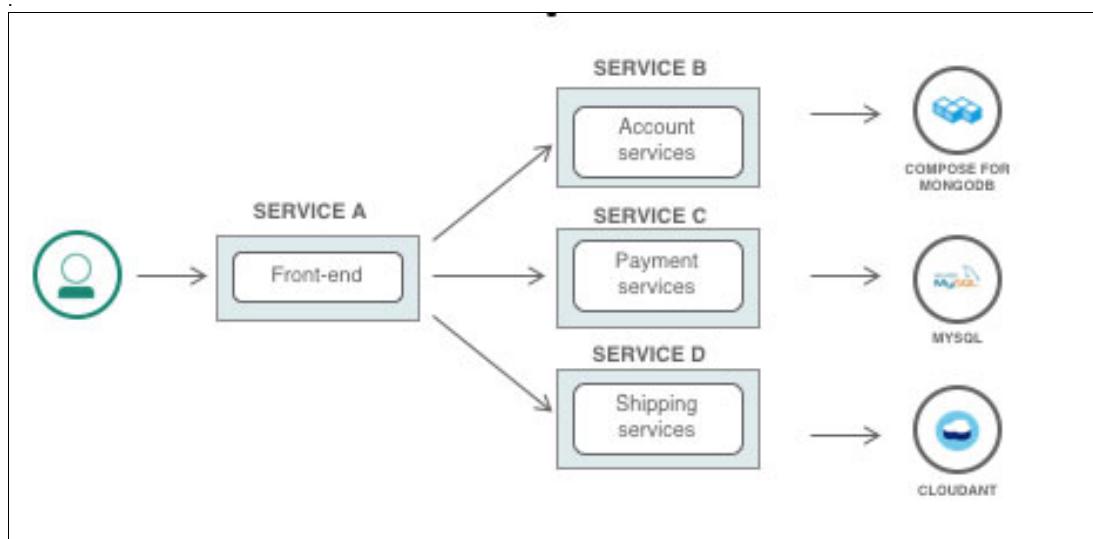


Figure 1-16 Microservices or cloud-native model

Key characteristics of cloud-native applications

Here are the key characteristics of a cloud-native application:

- Decentralized ownership: Development is typically distributed across multiple teams, with each team responsible for a specific microservice.
- Independent deployment: Microservices can be tested, updated, and scaled independently of one another.
- Service-to-service communication: All microservices within the application communicate by using lightweight protocols and APIs.
- Function separation: Each well-defined function is implemented as a dedicated microservice.

Advantages and disadvantages of cloud-native applications

Here are the advantages of cloud-native applications:

- ▶ Accelerated scalability: Cloud-native applications are suitable for large-scale systems. Multiple services can scale simultaneously based on demand.
- ▶ Automation-ready: Microservices support automation of the CI/CD pipeline, enabling frequent and reliable code updates.
- ▶ Faster deployment: Individual services can be deployed independently, allowing for quicker updates without redeploying the entire application.
- ▶ Increased cost efficiency: Resource allocation and maintenance are optimized by using defined services. Development efforts are concentrated, reducing overall operational costs.

Here are some of the disadvantages of cloud-native applications:

- ▶ Increased complexity: Managing communication between distributed microservices can be challenging. Developers might need to implement extra logic to help ensure smooth interactions between modules.
- ▶ Security vulnerabilities: Service-to-service communication typically relies on an API gateway, which can introduce security risks that are related to authentication and other critical processes.

Composite

Composite applications combine monolithic architectures with modern cloud-native components, enabling organizations to modernize gradually while preserving business-critical logic. This approach supports hybrid IT environments, where earlier systems interact with cloud services.

Composite applications integrate existing enterprise systems with modern APIs, allowing partial migration to cloud environments while retaining some earlier components. Middleware and API gateways facilitate communication between monolithic and cloud-native architectures, enabling greater agility through modular refactoring strategies.

These applications can aggregate multiple sources, services, or even entire applications into a single, unified interface presented to the user. In essence, they create modernized front ends for earlier systems. As a result, composite applications can be built by using any technology or architecture, as shown in Figure 1-17 on page 29.

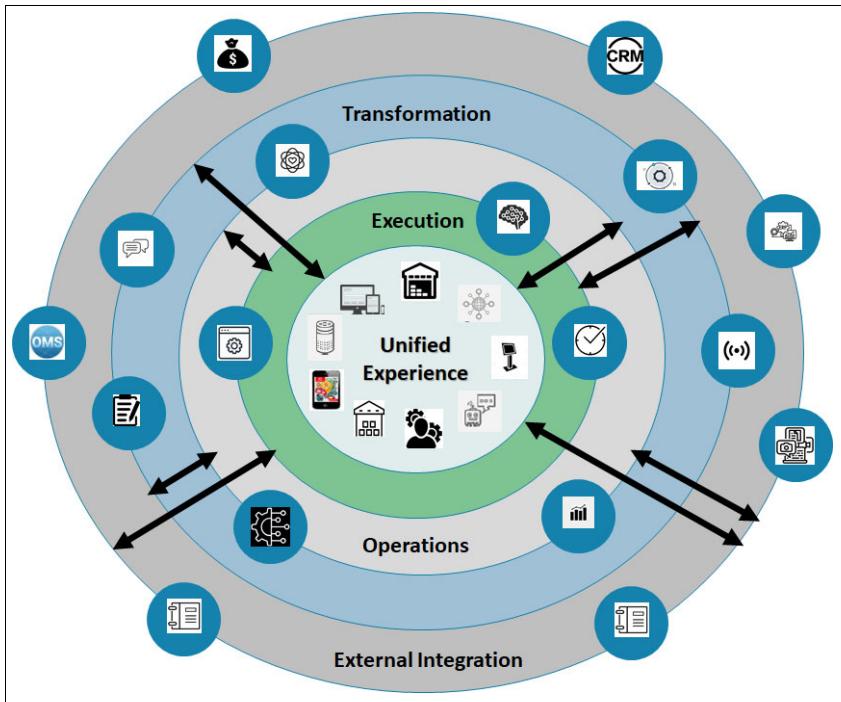


Figure 1-17 Composite application model

Key characteristics of composite applications

Here are the key characteristics of a composite application:

- ▶ Modular design: Composite applications are built by integrating various components (such as web services, databases, and applications) that can be developed and managed independently.
- ▶ External service integration: These applications rely heavily on accessing data and functions from external services.
- ▶ Orchestration layer: A middleware layer is typically used to manage interactions and data flow between components, helping ensure seamless integration and coordinated execution of tasks.

Advantages and disadvantages of composite applications

Here are the advantages of composite applications:

- ▶ Technology diversity: Composite applications can incorporate different technology stacks during development.
- ▶ Unified UI: The UI presents data from multiple systems through a single interface, eliminating the need to use separate applications.
- ▶ Improved data integration: These applications enable seamless access to data from various sources, including earlier systems, without requiring complex data transformation or manual entry.
- ▶ Customizable applications: Developers can create tailored solutions by combining different services to meet specific business requirements.

Here are some of the disadvantages of composite applications:

- ▶ Domino effect: If an underlying software component fails, the entire application might be affected due to its reliance on that component's availability and function.
- ▶ Security considerations: Managing security across multiple integrated systems can be complex and requires careful planning.
- ▶ Troubleshooting challenges: Identifying the root cause of issues can be difficult due to the interconnected nature of the components.

1.5.2 Virtualization technologies

Virtualization is the process of using software to create an abstraction layer over computer hardware, allowing the hardware elements of a single computer (or host machine) to be divided into multiple VMs.

A *hypervisor* is software that enables multiple VMs with each VM running their own OS to run on one physical server. The hypervisor pools allocate physical computing resources as needed by the VM, enabling efficiency, flexibility, and scalability. Hypervisors separate VMs from each other logically, assigning each its own part of the underlying computing power, memory, and storage. This approach prevents the VMs from interfering with each other. For example, if one OS suffers a crash or a security compromise, the others are unaffected.

A *VM* is an emulation of a physical computer. VMs allow multiple OSs to run on a single physical server, each within its own isolated environment. They interact with the host system through the hypervisor, which manages resource allocation and system integrity.

Containerization compared with virtualization

Containers are a foundational element of cloud-native applications, making it essential to understand what they are, and what they are not. Although containers and VMs have distinct characteristics, both improve efficiency, enhance application portability, and support the software development lifecycle.

Containers and VMs are two approaches to packaging computing environments that isolate them from the underlying system. The primary difference lies in what components are isolated, which affects their scalability and portability. Containers are a lightweight, agile alternative to VMs because they do not require a hypervisor. Instead of starting an entire VM, a container packages everything that is needed to run a single application or microservice, including its runtime libraries.

Containers use OS-level virtualization. They use features of the host OS to isolate processes and control access to CPU, memory, and storage resources. In most OSs, code runs in either user space or kernel space. Code in a kernel space has unrestricted access to system hardware, so a crash there can affect the entire system. In contrast, a crash in user space typically impacts only the affected application.

A container emulates the user space of an OS. Through container engines such as Docker, CoreOS, or CRI-O, the host OS exposes APIs that provide isolated processes, memory, file systems (mount points), and networking within this user space. Each container's user space is isolated from other applications, although all containers share the underlying hardware and OS kernel.

This architecture offers a secure and lightweight environment for applications. Containers start almost instantly because they do not require starting a full OS kernel (the host kernel is already running). However, this design introduces a single point of failure because all containers depend on the same kernel and hardware. This risk is mitigated through availability, scaling, and redundancy mechanisms that are provided by container orchestration platforms, such as Red Hat OpenShift Container Platform.

As a result, containers are ideally suited for microservices and stateless applications, where lightweight, scalable deployment is essential. In contrast, stateful applications, such as databases, are often better suited for VMs, which provide dedicated hardware abstraction and kernel isolation for stronger fault tolerance and resource control.

Although container technology has existed for decades, the modern container era began in 2013 with the introduction of Docker, an open-source platform for building, deploying, and managing containerized applications.

Unlike VMs, which virtualize hardware, containers virtualize the OS. Each container includes only the application and its dependencies, making containers small, fast, and portable. They do not require a guest OS, as they use the host OS's features and resources.

As part of a modernization journey on IBM Power, containers are preferred due to their lightweight architecture and portability. They enable faster deployment, greater consistency across environments, and simplified application lifecycle management, making them ideal for modern, cloud-native workloads on Power servers.

PowerVM

IBM PowerVM is the virtualization hypervisor that comes standard with IBM Power servers. It supports VMs by enabling the creation of LPARs and allowing resource sharing across multiple partitions. PowerVM is tightly integrated with IBM Power hardware, delivering enterprise-grade virtualization with minimal processing compared to other virtualization technologies.

PowerVM enables the consolidation of VMs running multiple workloads onto fewer systems, resulting in reduced costs, improved efficiency, faster deployment, enhanced workload security, better server usage, and a higher ROI.

The PHYP is the built-in hypervisor for IBM Power servers and is embedded in the system firmware. When a Power server is powered on, the PHYP is loaded along with the system firmware. This design minimizes virtualization processing for guest operating systems and provides the following hypervisor capabilities:

- ▶ IBM Micro-Partitioning®
- ▶ SPP
- ▶ VIOS
- ▶ Live Partition Mobility (LPM)
- ▶ Dynamic Logical Partitioning (DLPAR)
- ▶ Performance and capacity monitoring
- ▶ Capacity on Demand (CoD)
- ▶ Simplified Remote Restart (SRR)

The IBM Power Architecture® is *bi-Endian*, supporting both big-endian and little-endian platforms. As a result, the IBM Power Hypervisor supports AIX, IBM i, Linux, and CoreOS for Red Hat OpenShift guests. At the time of writing, Ubuntu is supported only on bare-metal Power servers.

Figure 1-18 shows the supported guest OSs on IBM PowerVM Hypervisor.

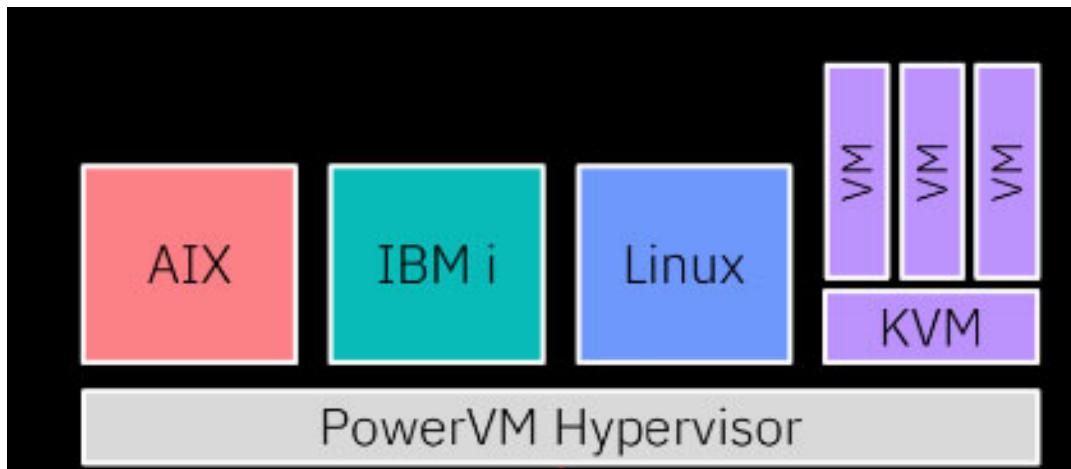


Figure 1-18 Supported guest operating systems on IBM PowerVM Hypervisor

PowerVM enables a single IBM Power server to host up to 1,000 VMs running a mix of operating systems and environments simultaneously. In addition to virtualization, PowerVM provides advanced features that help manage and control the virtualized environment effectively.

PowerVM virtualizes hardware resources, including processors, memory, storage I/O, and network I/O, allowing for flexible and efficient resource allocation. It also supports platform-level capabilities such as LPM and SRR.

Figure 1-19 shows an overview of PowerVM.

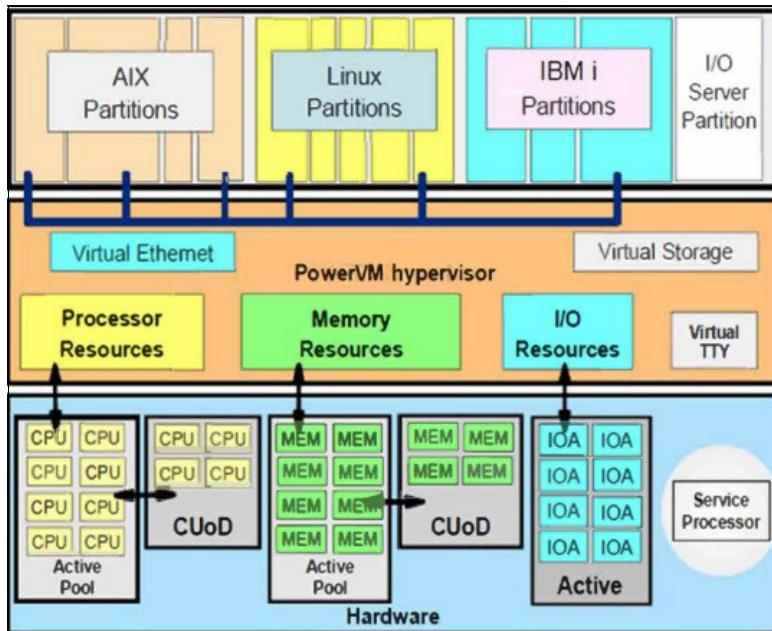


Figure 1-19 PowerVM overview

Micro-Partitioning

PowerVM allows a VM to initially use as little as 0.05 processing units, or 1/20 of a single processor core, and supports adjustments as small as 0.001 of a core. This level of granularity provides exceptional flexibility in aligning resource allocation with the precise needs of each workload.

With PowerVM Micro-Partitioning, you can allocate processor resources in increments as small as 0.001 of a CPU and provision those resources from targeted processor pools.

PowerVM Micro-Partitioning delivers flexibility and efficiency in allocating valuable computing assets, helping to optimize performance and reduce operational costs.

Figure 1-20 shows an overview of the architecture of multiple shared pools.

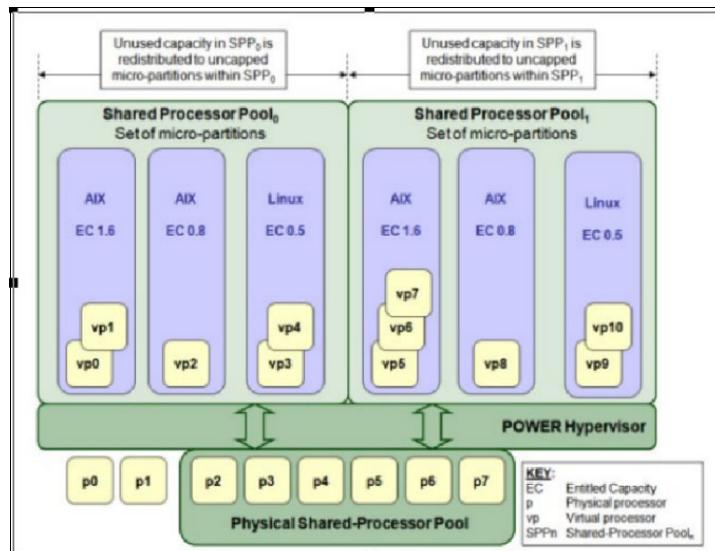


Figure 1-20 Overview of the architecture of multiple shared pools¹⁴

Shared Processor Pools

PowerVM enables effective usage of system resources by dynamically allocating only the processing capacity that is required by each partition. The hypervisor can automatically and continuously adjust the processing capacity that is assigned to each partition or VM based on system demand. You can configure a shared processor partition so that, if a VM requires more processing capacity than its assigned units, it can borrow unused capacity from the SPP.

Shared processors are physical processors whose capacity is distributed among multiple LPARs. The ability to divide and share physical processors across partitions is known as Micro-Partitioning technology.

The server distributes unused processing capacity among all uncapped shared processor partitions, regardless of the SPPs to which they are assigned. For example, if LPAR 1 is assigned to the default SPP and LPARs 2 and 3 are assigned to a different SPP, all three partitions still compete for the same pool of unused physical processor capacity on the server.

Note: For more information about shared processors, see [Shared processors](#).

¹⁴ Source: <https://www.redbooks.ibm.com/technotes/tips1091.pdf>

Virtual I/O Server

VIOS enables the sharing of storage and network resources across multiple VMs simultaneously. This capability helps reduce costs by allowing the precise configuration of hardware resources based on system requirements.

VIOS is a software component that runs within an LPAR and is a required feature of the PowerVM Editions hardware offering. It provisions the sharing of physical I/O resources among client LPARs within an IBM Power Managed System. VIOS provides virtual SCSI (vSCSI) targets, virtual Fibre Channel, and Shared Ethernet Adapter (SEA) functions to client LPARs within the server.

Figure 1-21 shows an example of the VIOS architecture.

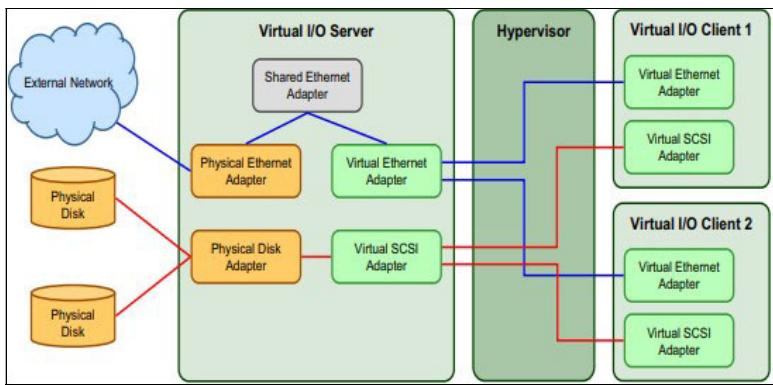


Figure 1-21 Architecture view of the VIOS

A best practice for the deployment of the VIOS is in a dual-VIOS configuration on each IBM Power server.

Note: For more information about VIOS, see [Virtual I/O Server overview](#).

Virtual I/O Server storage support

The VIOS provides many options for sharing storage across multiple partitions:

- ▶ vSCSI

When using vSCSI, storage adapters and volumes are assigned to the VIOS, and vSCSI adapters are defined to connect those devices to the client LPARs. These vSCSI adapters can share disk storage, tape drives, or optical devices that are assigned to the VIOS LPAR. All storage device configurations remain on the VIOS, simplifying the setup and management of the client LPARs.

Figure 1-22 on page 35 shows a standard vSCSI configuration for a managed system.

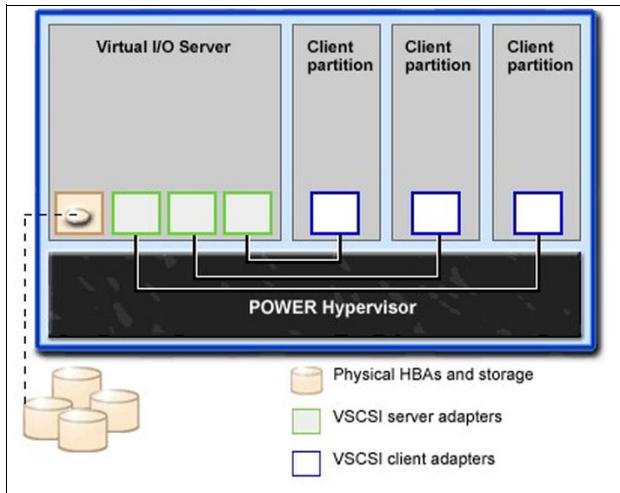


Figure 1-22 Standard vSCSI configuration for a Managed System

Note: For more information about vSCSI storage, see [Virtual SCSI](#).

- ▶ Shared Storage Pool (SSP)

An SSP is an extension of the vSCSI connection, in which the storage devices that are assigned to the VIOS are configured into a shared pool. This configuration further simplifies storage management in a PowerVM environment.

An SSP is a collection of storage resources that can be shared across multiple VMs or LPARs. It provides a unified and flexible storage infrastructure by abstracting physical storage resources. As a result, guest operating systems within LPARs do not need to be aware of the underlying physical storage. Each VM accesses the storage pool based on defined policies, such as storage allocation, data replication, or redundancy management.

Multiple physical disks can be pooled to create a virtualized storage system, eliminating the need for individual disk management. VMs can dynamically request or release storage from the pool based on workload demands, improving resource usage. The VIOS manages these pools, allowing VMs or LPARs to access storage transparently.

If a disk or storage unit fails, the SSP can provide failover mechanisms to help ensure continued VM operation with minimal downtime. Storage within the pool can be resized, expanded, or contracted to adapt to changing requirements.

With SSP, storage is used more efficiently because it can be dynamically allocated where needed, avoiding wasted capacity. New storage can be added to the pool without downtime, and unused storage from one VM can be reassigned to others, improving overall system performance and usage.

Figure 1-23 shows a standard SSP architecture.

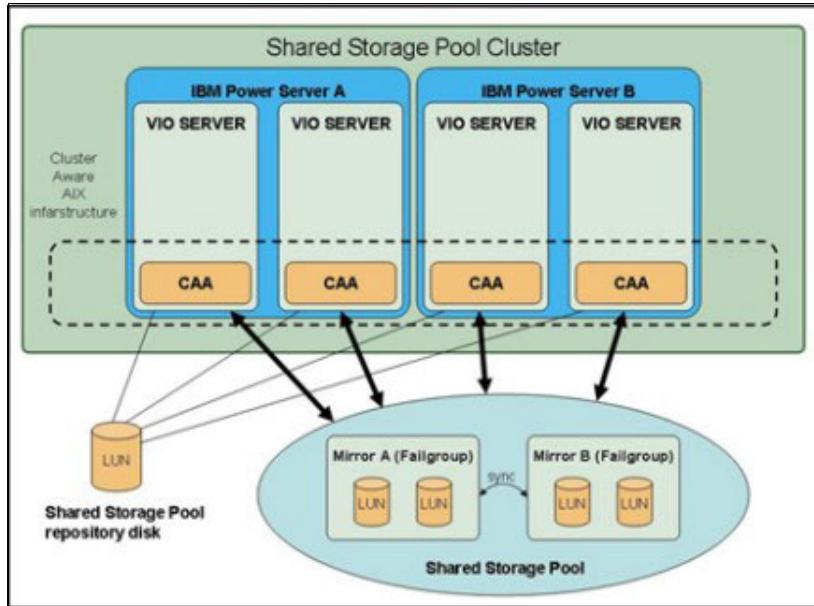


Figure 1-23 Standard SSP architecture

- ▶ N_Port ID Virtualization (NPIV)

Using NPIV allows virtual FC devices to pass through the VIOS directly to the client partition through virtual FC adapters. Logical Unit Numbers (LUNs) are zoned directly from the storage area network (SAN) to the partition's virtual FC adapter. This technology enables multiple LPARs to access independent physical storage through the same physical FC adapter port on the VIOS while maintaining zoning security within the FC infrastructure.

Figure 1-24 shows NPIV for FC device support.

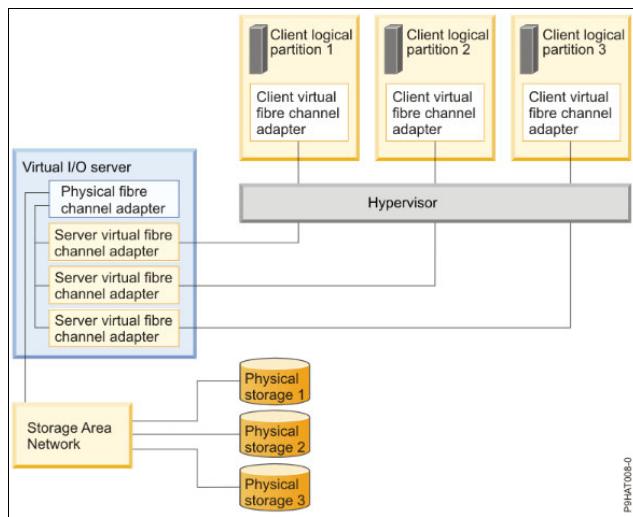


Figure 1-24 NPIV for Fibre Channel device support

As a best practice for production environments, use FC devices with NPIV in a dual-VIOS configuration to help ensure SAN storage volume redundancy.

- ▶ FC-Nonvolatile Memory Express (NVMe)

New enhancements in NPIV on IBM Power10 servers introduce support for the NVMe protocol over FC fabrics. NVMe over FC improves data transfer speeds, enabling faster access between hosts and storage systems.

Figure 1-25 illustrates the performance advantages of NVMe compared to conventional SCSI-based devices.

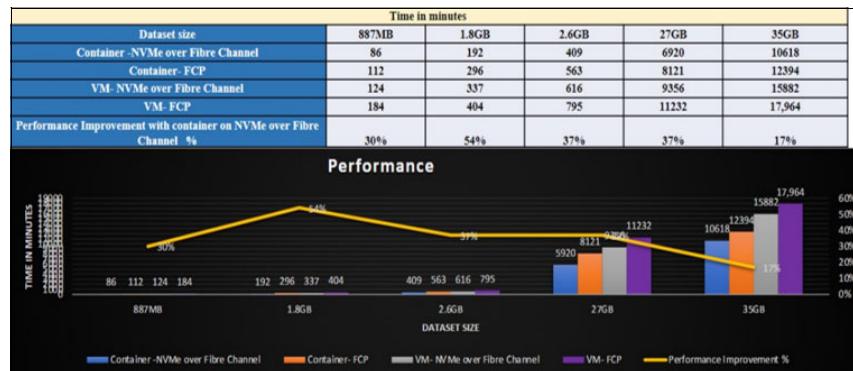


Figure 1-25 NVMe performance

NVMe over FC produces more efficient I/O performance for AI workloads when compared to traditional FCP and SCSI protocols. NVMe over FC enables faster access between hosts and storage systems with a lighter and leaner driver stack that runs faster and uses fewer resources.

NVMe over FC demonstrates more efficient I/O performance for AI workloads compared to traditional FCP and SCSI protocols. It enables faster access between hosts and storage systems through a lightweight driver stack that operates more efficiently and uses fewer system resources.

Note: For more information about containerized ML models, see the [BrightTalk presentation on the Benefits of FC-NVMe](#).

The PowerVM Virtual Switch

PowerVM enables the definition of virtual Ethernet adapters, allowing client partitions to send and receive network traffic without requiring a dedicated physical Ethernet adapter. A virtual Ethernet adapter can also facilitate communication between LPARs within the same system, eliminating the need for physical Ethernet adapters.

Within the system, virtual Ethernet adapters connect to an IEEE 802.1Q virtual Ethernet switch. The default switch name is ETHERNET0, but PowerVM allows the creation of more virtual switches based on customer requirements.

By default, the PowerVM virtual switch operates in Virtual Ethernet Bridge (VEB) mode. It can also operate in Virtual Ethernet Port Aggregator (VEPA) mode. Using the PowerVM virtual switch, LPARs can communicate through virtual Ethernet adapters by assigning tagged VIDs. These VIDs allow virtual Ethernet adapters to share a common logical network. The system transmits packets by copying them directly from the sender LPAR's memory to the receiver LPAR's receive buffers, avoiding intermediate packet buffering.

When the PHYP switch operates in VEB mode, VMs on the same tagged VID can use the hypervisor virtual switch for high-performance network frame relays. When using non-bridged PVIDs, network packets remain within the hypervisor, enabling direct in-memory transfers.

VEPA mode, which is part of the IEEE 802.1Qbg standard, offloads the complexity and performance demands of hypervisor-based virtual switch bridging. In this mode, all frame relay switching is handled by the external departmental switch, which then manages guest port-to-port communication, and network services such as firewalls, access control lists (ACLs), quality of service (QoS), and port mirroring.

When VM-to-VM traffic is routed through the enterprise switch for firewall filtering, the network packet management load shifts away from the hypervisor, which reduces the hypervisor load and might improve performance for partitions with high network usage.

Live Partition Mobility

LPM enables the migration of running VMs across different physical systems without disrupting the operating systems or applications running within them.

By using LPM, you can move running AIX, IBM i, and Linux LPARs (including their operating systems and applications) from one system to another without requiring shutdown. LPM capabilities are integrated into the PHYP and are enabled through a licensed feature. A best practice implementation of PowerVM is required for successful deployment.

To help ensure storage access for a partition that is moved between two managed systems, the SAN LUNs must be zoned and masked for access through both systems.

DLPAR operations

DLPAR operations enable the dynamic allocation of extra resources (such as available processor cores and memory) to a VM without stopping the application.

Here are some of the DLPAR operations:

► **Performance and Capacity Monitoring**

Supports the collection of key performance statistics, providing administrators with insights into physical resource distribution among VMs. Continuous monitoring helps ensure that resources are evenly distributed and optimally used.

► **CoD**

IBM Power servers are delivered with processors in a combination of the following states:

- Static: Delivered and dedicated to the PowerVM managed system.
- Mobile: Can be moved between PowerVM managed systems.
- Inactive (Dark): Installed but not activated; available for future activation as either static or mobile.

CoD offerings enable you to dynamically activate one or more resources on your server based on business needs. You can temporarily or permanently activate inactive processor cores or memory units that are already installed on the system.

Note: For more information about CoD, see the following resources:

► [Capacity on Demand](#)

► [Elastic Capacity on Demand - Overview and specifications](#)

- ▶ Remote Restart

Remote Restart enables rapid recovery by allowing a VM to restart on a different physical server when an error causes an outage.

SRR is a supported, configurable HA option for LPARs on Power10 processor-based servers. When a server outage occurs, SRR allows the affected LPAR to restart on another physical server. CoD entitlements can be used to dynamically allocate system resources on the target server. SRR is also used in the AIX Live Update process.

Kernel-based Virtual Machine

Kernel-based Virtual Machine (KVM) is an open-source, full virtualization technology for Linux operating systems. With KVM, Linux functions as a hypervisor that runs multiple, isolated VMs. In the KVM architecture, each guest (VM) is implemented as a standard Linux process.

After KVM is installed, you can run multiple guest VMs, each with its own operating system image. Each VM includes private, virtualized hardware components such as memory, storage, and a network card.

KVM in a PowerVM LPAR

KVM is an extra virtualization option that is available on Power10 servers that run PowerVM. It brings the performance, flexibility, and efficiency of KVM virtualization technology to a PowerVM LPAR. An LPAR running a KVM-enabled Linux distribution can host PPC64-LE KVM guest VMs. These KVM guests use the existing resources that are assigned to the LPAR.

Figure 1-26 shows KVM support in a PowerVM LPAR.

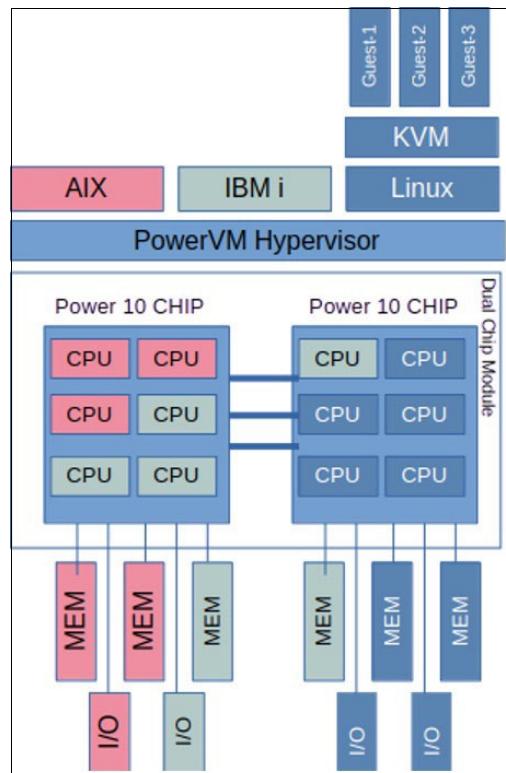


Figure 1-26 PowerVM with KVM support

Software levels

To enable KVM in a Power10 LPAR, the following code levels and Linux distributions are required:

- ▶ Firmware level: FW1060.10
- ▶ Hardware Management Console (HMC): V10 R3 SP1060 or later

KVM is enabled in Linux kernel 6.8 and Quick Emulator (QEMU) 8.2, and is supported on the following Linux distributions:

- ▶ Fedora 40 with kernel 6.10
- ▶ Ubuntu 24.04

Industry-standard Linux Virtualization Stack

KVM in a PowerVM LPAR uses the industry-standard Linux KVM virtualization stack and can integrate with an existing Linux virtualization ecosystem.

Figure 1-27 shows the industry-standard Linux virtualization stack.

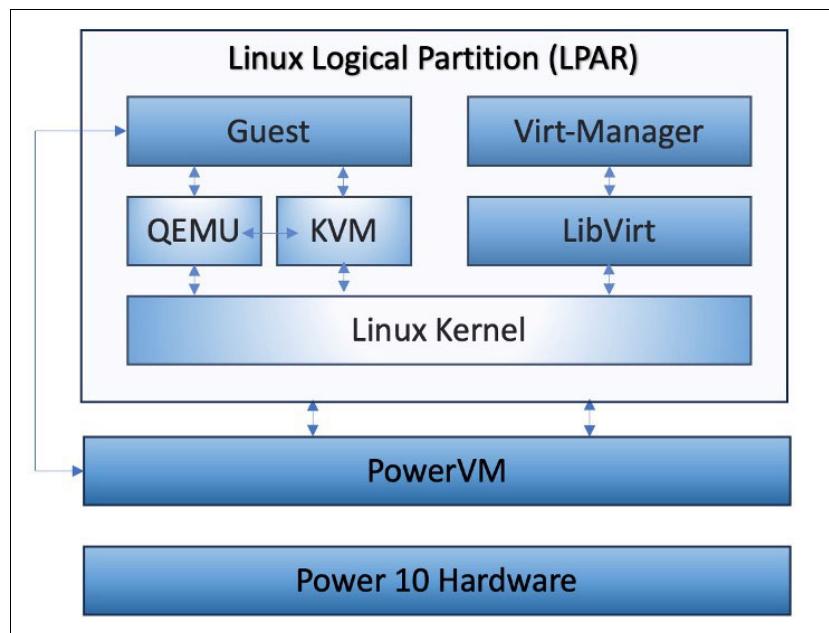


Figure 1-27 Industry-standard Linux Virtualization Stack

KVM in an LPAR is enabled through the following components:

- ▶ IBM Power Architecture and Power10

Power10 servers offer advanced virtualization capabilities that allow multiple operating system instances to share hardware resources while maintaining isolation. The Radix Memory Management Unit architecture enables independent management of page tables for both the LPAR and its KVM guest instances.

- ▶ PowerVM

PowerVM, the industry-leading virtualization stack, introduces new functions to create and manage KVM guests. These enhancements extend the Power platform architecture by adding new hypervisor interfaces.

- ▶ Linux kernel with KVM module

The Linux kernel includes the KVM module, which provides the core virtualization infrastructure for running multiple VMs within a Linux host LPAR. Upstream kernels and supported downstream distributions, such as Fedora and Ubuntu, use new Power Architecture extensions to create and manage KVM guests in Linux on Power LPARs.

- ▶ QEMU

QEMU is the user-space component that implements VMs on the host by using KVM functions.

- ▶ Libvirt

Libvirt provides a toolkit for managing VMs.

For more information, see [KVM in a PowerVM LPAR](#).

1.5.3 Containerization solutions on Power

Containers are executable units of software that package application code along with its libraries and dependencies. They offer a lightweight, agile approach to virtualization by eliminating the need for a hypervisor.

Instead of virtualizing the underlying hardware, containers virtualize the operating system (typically Linux or Windows). Each container includes only the application and its required libraries and dependencies. Containers are small, fast, and portable because they do not include a guest operating system. Instead, they use the features and resources of the host OS.

Like VMs, containers help improve CPU and memory usage on physical systems. However, containers go further by enabling microservice architectures, where application components can be deployed and scaled independently. This approach offers a more efficient alternative to scaling an entire monolithic application when only a single component experiences increased load.

Kubernetes

Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. It provides a framework for efficiently managing the complexities of running applications in containers across a cluster of machines.

Here are the key components of a Kubernetes cluster:

- ▶ Control plane

The control plane is the central management unit of a Kubernetes cluster.

- ▶ Worker nodes

Worker nodes (worker machines or servers) run the containerized applications and workloads.

- ▶ Pods

Pods are the fundamental building blocks in Kubernetes. Each pod groups one or more containers and provides a shared environment, including network and storage resources.

- ▶ Controllers

Controllers help ensure that the cluster's actual state matches the wanted state.

- ▶ Services
Services enable communication and load balancing between different sets of pods. They make applications discoverable and resilient.
- ▶ Volumes
Volumes provide persistent storage to containers within pods.
- ▶ ConfigMaps
ConfigMaps stores non-sensitive configuration data as key-value pairs.
- ▶ Secrets
Secrets are used to store sensitive information.
- ▶ Namespaces
Namespaces organize and partition resources within a cluster.
- ▶ Ingress
Ingress is a resource that manages external access to services within the cluster.

Figure 1-28 shows the components of a Kubernetes cluster.

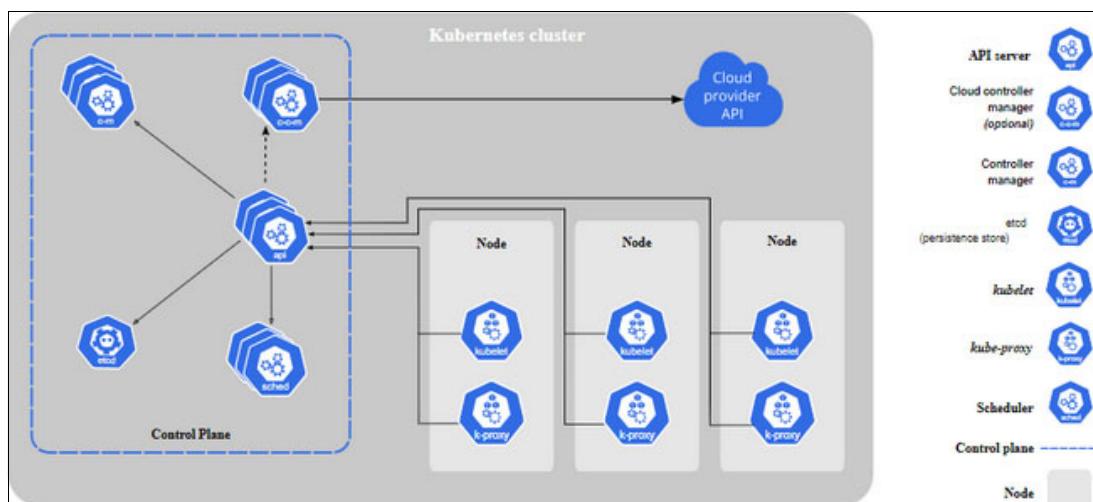


Figure 1-28 Components of a Kubernetes cluster

Red Hat OpenShift

Red Hat OpenShift is a leading enterprise Kubernetes platform that provides a robust foundation for developing, deploying, and scaling cloud-native applications. It extends Kubernetes with more features and tools to enhance productivity and security, making it an ideal choice for organizations that want to use container technology at scale.

Red Hat OpenShift is a unified platform to build, modernize, and deploy applications efficiently. It offers a complete set of services to accelerate application delivery across your choice of infrastructure, whether public cloud, on-premises, hybrid cloud, or edge environments. Red Hat OpenShift delivers a consistent experience across all deployment models.

Red Hat OpenShift provides a flexible platform that supports a wide range of business needs, from an enterprise-grade Kubernetes orchestrator to a comprehensive cloud-native application development environment that can be deployed as a self-managed solution or used as a fully managed cloud service.

Here are the key capabilities of Red Hat OpenShift:

- ▶ Deploy and run applications in any environment. Red Hat OpenShift supports building new applications, modernizing existing ones, running third-party independent software vendor (ISV) applications, and by using public cloud services, all under a single platform.
- ▶ Integrate data analytics, and AI and ML capabilities into cloud-native applications to deliver deeper insights and greater value.
- ▶ Deploy and manage containerized workloads consistently across environments. Red Hat OpenShift helps infrastructure investments become future-ready and delivers speed and flexibility on-premises across cloud environments to the edge.
- ▶ Gain end-to-end management, observability, and security across the entire architecture. Red Hat OpenShift includes enterprise-grade security features to meet compliance requirements.

Built by open-source leaders, Red Hat OpenShift includes an enterprise-ready Kubernetes solution with a choice of deployment and usage options to meet the needs of your organization. From self-managed to fully managed cloud services, you can deploy the platform in the data center, in cloud environments, and at the edge of the network.

With Red Hat OpenShift, you can get advanced security and compliance capability, end-to-end management and observability, and cluster data management and cloud-native data services, such as the following ones:

- ▶ Red Hat Advanced Cluster Security for Kubernetes: Modernizes container and Kubernetes security so that developers can add security controls early in the software lifecycle.
- ▶ Red Hat Advanced Cluster Management for Kubernetes: You can manage your entire application lifecycle and deploy applications on specific clusters based on labels.
- ▶ Red Hat OpenShift Data Foundation: Supports performance at scale for data-intensive workloads.

AIX Workload Partitions

An early implementation of container-like technology was developed by IBM for the AIX operating system. Introduced with AIX 6.1 in 2007, AIX Workload Partitions (WPARs) provided a form of operating system-level virtualization that enabled application isolation and resource control for multiple workloads running simultaneously.

Predating container technologies such as Docker (introduced in 2013), AIX WPARs allowed a single AIX server to host multiple virtualized AIX environments (WPARs) within a single AIX image that owned all the physical system resources (referred to as the global environment). WPARs also supported Versioned WPARs, which allowed older versions of AIX (such as AIX 5.2 and 5.3) to run within newer environments. This capability enabled clients to migrate earlier applications to later IBM Power hardware while maintaining compatibility.

AIX WPARs supported live application mobility, allowing a WPAR to be moved from one LPAR or physical server to another without stopping the application that runs inside it.

Although still supported in AIX 7.3 as of December 2024, the usage of WPARs declined over time. Also, AIX 6.1 is no longer supported within a WPAR environment.

Note: For more information about AIX WPARs, see [AIX](#).

1.5.4 Automation

IBM and IBM Business Partners offer various automation solutions for IBM Power servers, ranging from Ansible to IBM Turbonomic®:

- ▶ Ansible Collections

IBM Power products include several Ansible collections that are designed to automate tasks and streamline operations across all management layers, from hardware to applications. Ansible collections are available for every management interface that is available on IBM Power, from the hardware level (HMC and VIOS) to the OS level, and even for applications (SAP HANA) and databases (Oracle).

Here are some key collections:

- AIX Collection

Automates tasks such as patching, user and group management, boot management, command execution, and object authority management. It integrates with familiar AIX tools like Network Installation Manager (NIM) and `alt_disk_copy`.

- IBM i Collection

Provides modules for patching, user and group management, boot management, SQL query execution, and more.

- Linux Collection

Supports automation for Linux environments that run on IBM Power servers.

- VIOS Collection

Focuses on managing VIOSSs, including creation, installation, and configuration.

- HMC Collection

Includes modules for fix management, LPAR management, Power server management, password policy configuration, and dynamic inventory generation.

These collections help administrators efficiently manage IBM Power environments and integrate them into broader automation strategies.

- ▶ IBM Turbonomic

IBM Turbonomic automates the monitoring and performance management of IBM Power servers. It helps organizations optimize compute, network, and storage resources to maximize ROI.

Turbonomic provides a comprehensive view of resource usage and continuously automates the most efficient allocation of resources at the lowest possible cost. It supports any application stack, whether it is deployed on-premises, in the cloud, or in hybrid environments.

Turbonomic can understand any application stack, whether it is in an on-premises data center, cloud, or hybrid cloud environment. By automating resource allocation decisions, Turbonomic reduces the operational burden on IT teams, allowing them to focus on organizational tasks. It also enables cost comparisons for running workloads across different environments by analyzing the resource requirements of business applications.

Note: For more information about Turbonomic, see this [IBM product website on Turbonomic](#).

Managing Red Hat OpenShift Clusters

You can manage your Red Hat OpenShift clusters by using either Red Hat Advanced Cluster Management for Kubernetes or IBM Cloud Pak for Multicloud Management. Both solutions support a combination of console-based and command-line interface (CLI) operations.

Red Hat Advanced Cluster Management for Kubernetes

Red Hat Advanced Cluster Management for Kubernetes addresses common challenges that administrators and Site Reliability Engineers (SREs) encounter. It provides centralized visibility and control over clusters and applications (whether containerized or virtualized) through a unified console. Preconfigured governance policies can be applied consistently across environments, enabling users to manage operations from anywhere on Red Hat OpenShift and to oversee other supported Kubernetes clusters within their fleet.

Red Hat Advanced Cluster Management extends its multi-cluster management capabilities to environments built on the Power Architecture. Specifically, it supports the management of Red Hat OpenShift clusters deployed on IBM Power servers, offering centralized control and visibility alongside other Kubernetes deployments. Organizations may apply consistent policies, streamline application lifecycles, and monitor cluster health across diverse infrastructure, including IBM Power processor-based servers.

Red Hat Advanced Cluster Management supports the creation of Red Hat OpenShift clusters on the following platforms:

- ▶ Virtualization Platforms:
 - KVM
 - VMware ESXi
 - Nutanix AHV
 - IBM PowerVM
 - IBM z/VM®
- ▶ Public Cloud Providers:
 - AWS
 - Microsoft Azure
 - Google Cloud Platform (GCP)
 - Microsoft Azure Government
 - AWS GovCloud
 - Alibaba Cloud
 - Oracle Cloud Infrastructure
 - IBM Cloud

For a full list of supported deployment options for Red Hat OpenShift, see this [documentation](#).

In summary, Red Hat Advanced Cluster Management simplifies and automates the management of Kubernetes environments, particularly ones that span multiple locations or cloud providers.

For more information about Red Hat Advanced Cluster Management, see the [Red Hat Advanced Cluster Management data sheet](#).

IBM Cloud Pak for MultiCloud Management

With the IBM Cloud Pak for Multicloud Management console and CLI tools, you can view cluster information, manage cluster labels, and access metering usage data for the hub cluster only.

IBM Cloud Pak for Multicloud Management is deployed within a Red Hat OpenShift cluster, which serves as the hub cluster. This hub cluster provides a unified view of both cloud and on-premises resources that interact with it. The platform extends visibility beyond containerized environments to include traditional workloads, such as VMs within virtualization clusters and stand-alone VMs, even workloads outside the Red Hat OpenShift environment.

This comprehensive visibility enhances operational awareness and enables automation across interconnected tasks throughout the entire infrastructure.

Figure 1-29 shows how IBM Cloud Pak for MultiCloud Management is deployed.

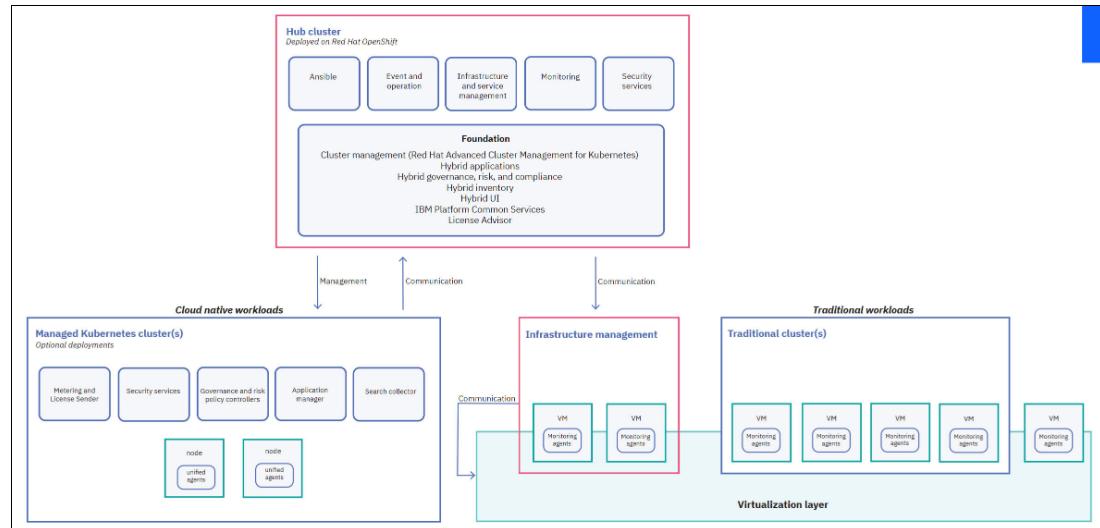


Figure 1-29 IBM Cloud Pak for MultiCloud management deployment architecture

Figure 1-30 shows the supported Red Hat OpenShift versions in IBM Cloud Pak for Multicloud Management 2.3 Fix Pack 10 at the time of writing.

Platform	OpenShift Container Platform version
Linux® x86_64	4.14, 4.16, 4.18
Linux® on Power® (ppc64le)	4.14, 4.16, 4.18

Figure 1-30 Supported Red Hat OpenShift versions

In addition to its default features for managing multicloud environments, IBM Cloud Pak for Multicloud Management includes several installable modules that can be added to your cluster to enhance application and infrastructure management and automate operational tasks:

- ▶ **Monitoring Module**
Enables monitoring of performance and availability for cloud applications across hybrid cloud environments.
- ▶ **Terraform and Service Automation Module**
Supports cluster security, operational efficiency, and consistent service level delivery.

- ▶ Infrastructure Manager
Provides centralized control and management of cloud infrastructure resources.
- ▶ Red Hat Ansible Automation
Integrates Ansible to run automation tasks across your environment.

Note: For more information, see [What's new in IBM Cloud Pak for Multicloud management](#).

1.6 IBM Power and artificial intelligence

The usage of AI is projected to unlock nearly \$16 trillion in productivity gains by 2030.¹⁵ Today's customers expect seamless experiences and timely responses to their inquiries. Organizations that fail to meet these expectations risk falling behind.

Investment in generative AI is expected to grow by a factor of four over the next 2 - 3 years. However, it still represents a small portion of total AI spending.¹⁶ Notably, 89% of enterprise decision-makers agree that scaling AI provides a competitive advantage.¹⁷

As AI (particularly generative AI) moves from ideation to operationalization, enterprises are seeking infrastructure that is reliable, hybrid-ready, and capable of delivering trusted insights. IBM Power offers an accelerated, flexible, and secure platform that is designed for enterprise AI workloads. IBM Power clients also benefit from having valuable enterprise data already on their IBM Power servers, enabling them to derive trusted insights and fully realize the benefits of AI.

1.6.1 Why artificial intelligence on IBM Power

IBM Power provides a trusted foundation to support your AI strategy. IBM Power servers are suitable for AI workloads due to their high performance, built-in AI acceleration, and robust security features. They integrate seamlessly with hybrid cloud environments and AI platforms such as IBM Watsonx®, making them a versatile and efficient choice for deploying AI applications at scale.

Accelerating efficiently

AI-optimized hardware and software enable clients to accelerate AI workloads efficiently (without requiring data scientists to modify their code) by delivering available performance optimization.

Improved performance

IBM Power10 hardware includes features optimized for AI workloads, such as the Matrix Math Accelerator (MMA). Combined with the large memory capacity and high parallelism of IBM Power10 servers, these capabilities deliver efficient and cost-effective acceleration for AI tasks.

For large language models (LLMs), IBM Power10 demonstrates significant performance advantages. For example, IBM Power S1022 servers can process up to 42% more batch queries per second than comparable x86 servers under peak loads of 40 concurrent users, while maintaining inferencing latency below 1 second.¹⁸

¹⁵ [Fortune, April 20, 2023: IBM CEO: 'Today's workforce should prepare to work hand in hand with AI'](#)

¹⁶ [IBM Institute for Business Value, Generative AI: The state of the market](#)

¹⁷ [Forrester Consulting Thought Leadership Paper: Overcome Obstacles To Get To AI At Scale](#)

¹⁸ For more information, see "Improved performance" on page 430.

Running AI on a highly performant sustainable platform

IBM Power10 enhances sustainability by delivering 39% more inferencing performance per watt compared to Intel-based servers.¹⁹

Improved economics

Clients can use the parallel inferencing capabilities and higher usage of the IBM Power platform to achieve 51% lower TCO over a 3-year period when running parallel inferencing in CP4D on Power S1022 compared to a similar x86 server.²⁰

1.6.2 Orchestrating AI flexibly

IBM Power provides clients the choice to create and run their AI workloads where and how needed by providing:

- ▶ A frictionless hybrid infrastructure that is built to be consistent at all layers – infrastructure, operating-system, virtualization, and software - whether on-premises, in a private/managed cloud or in the public cloud.
- ▶ A flexible consumption model with pay-as-you-use licensing for infrastructure and platform software regardless of where the workload is being run.
- ▶ A combination of enterprise and open-source software for AI providing the choice of building blocks for creating best fit AI workloads to serve their business needs.

1.6.3 Safeguarding AI and data

Enterprise clients are increasingly concerned with safety, risk, vulnerabilities, and compliance, especially as AI models process sensitive data at scale. To address these concerns, data must be protected through robust governance and security mechanisms:

- ▶ The IBM Power10 server simplifies encryption and supports end-to-end security through transparent memory encryption, which uses hardware-based features to deliver a seamless user experience without impacting performance.
- ▶ To minimize latency and consolidate cryptographic operations, IBM Power10 includes on-chip cryptographic algorithm acceleration so that algorithms such as Advanced Encryption Standard (AES), SHA-2, and SHA-3 run efficiently without offloading data to external accelerators.
- ▶ IBM Power10 also enhances application and data protection through secure virtual machine (VM) isolation, offering fewer Common Vulnerabilities and Exposures (CVEs) compared to hypervisors on x86-based servers.
- ▶ For compliance and audit readiness, IBM PowerSC provides centralized capabilities to manage, monitor, report, and visualize security and compliance. These features support regulatory requirements, including GDPR, through real-time updates and predefined security compliance profiles.

1.6.4 Hybrid flexibility

Hybrid flexibility is essential for deploying AI workloads. IBM Power servers deliver this flexibility by enabling enterprises to run AI both on-premises and in the cloud through IBM Power Virtual Server.

¹⁹ For more information, see “Running AI on a highly performant sustainable platform” on page 430.

²⁰ For more information, see “Improved economics” on page 430.

In addition to infrastructure flexibility, choice matters at higher levels of the AI solution stack. IBM Power supports a wide range of AI-optimized software options:

- ▶ Enterprise software
- ▶ Open-source, community-supported software
- ▶ Open-source, enterprise-supported software

1.6.5 Artificial intelligence solutions on IBM Power

Here are some solutions that enable AI on IBM Power servers.

IBM Cloud Pak for Data

IBM Cloud Pak for Data (CP4D) is a modular platform that integrates a broad set of software components for data analysis, organization, and management. As part of IBM's enterprise AI solutions for IBM Power, CP4D includes a wide range of components from Watson, Apache, IBM Db2, and Red Hat, which are all designed to accelerate data analytics tasks. As the platform continues to evolve, more capabilities and services within CP4D become available on IBM Power servers.

Open-source solutions

IBM Power supports both community and enterprise-backed open-source AI solutions. These capabilities are delivered through RocketCE and the Rocket AI Hub for IBM Power:

- ▶ RocketCE is a curated packaging of open-source AI tools that are optimized for IBM Power10 by using its on-chip acceleration features. It is available through the Rocket Software public Anaconda channel.²¹
- ▶ Rocket AI Hub for IBM Power is an integrated, freely available set of best in class, open-source AI platform tools, including Katib, Kubeflow, Kubeflow Pipelines, KServe, and RocketCE. All tools are delivered as container images and are designed to run in Kubernetes-based environments, such as Red Hat OpenShift.

These tools are integrated through Kubeflow and optimized to use the unique AI hardware capabilities of the IBM Power platform, enabling efficient and scalable AI development and deployment.

IBM watsonx

As the market increasingly adopts foundation models for generative AI use cases, the IBM Power platform is well positioned to deliver these capabilities through integration with IBM watsonx. The IBM Power platform is designed to support the inferencing of foundation models, enabling clients to deploy generative AI solutions that enhance customer experiences, boost productivity, and optimize business processes.

Generative AI workloads benefit from IBM Power10 on-chip acceleration, which delivers a differentiated and efficient experience for enterprise clients. As compute demands evolve, the IBM Power platform remains committed to providing a platform that is cost-effective, sustainable, resilient, and secure.

²¹ Source: <https://anaconda.org/rocketce/repo>

Red Hat OpenShift

In addition to supporting AI workloads for IBM clients, advancements in Red Hat OpenShift are influencing AI solution architectures. One such advancement is the introduction of Multi-Architecture Cluster support. This capability enables clients to deploy a Red Hat OpenShift cluster with both x86 and IBM Power compute nodes. It allows workloads to be deployed where they are best suited by using the strengths of each architecture and enhancing deployment flexibility.

IBM Power clients now have access to a comprehensive suite of AI capabilities that take advantage of IBM Power10 on-chip acceleration. These capabilities support both enterprise and open-source AI solutions and are aligned with key market drivers. With these tools, IBM clients can address critical business challenges by extracting actionable insights from their growing volumes of multi-modal data.



Modernization considerations

Modernizing applications and infrastructure is a complex process that requires careful consideration of numerous factors. As you modernize your environment, focus on your key business requirements. Maintaining a secure and reliable infrastructure while preserving flexibility and agility is essential. This approach enables your infrastructure to adapt quickly to new challenges and generate greater value from existing data assets.

Here are the factors that you should consider:

- ▶ Help ensure that all modernization initiatives align with overarching business goals.
- ▶ Conduct thorough assessments of existing infrastructure and applications.
- ▶ Implement modernization in a phased manner to minimize disruption.
- ▶ Continuously optimize costs through efficient resource usage.
- ▶ Automate processes to improve efficiency and reduce human error.
- ▶ Implement comprehensive monitoring and observability solutions.
- ▶ Help ensure compliance with all relevant regulations and industry standards.

Modernization efforts must address various factors, including security, storage, open-source technologies, and high availability (HA). A well-rounded strategy that incorporates these elements can help your organization unlock the full potential of the IBM Power environment, ensuring that it remains resilient, scalable, and secure in the digital age.

Also, integrating artificial intelligence (AI) and code assistants into your modernization strategy offers significant advantages. These tools streamline automation, boost development productivity, maintain high code quality, and optimize infrastructure management. With AI-driven insights and the capabilities of code assistants, you can enhance the speed, efficiency, and security of your applications while maintaining the flexibility and scalability needed to thrive. As technology evolves, adopting these tools will be essential to remain competitive in an increasingly fast-paced digital landscape.

The following topics are described in this chapter:

- ▶ 2.1, “Overview” on page 52
- ▶ 2.2, “Security” on page 53
- ▶ 2.3, “Open-source software on IBM Power” on page 57
- ▶ 2.4, “Storage options” on page 59
- ▶ 2.5, “High availability and disaster recovery” on page 72
- ▶ 2.6, “AI code assistants” on page 80

2.1 Overview

When modernizing applications and infrastructure, consider several critical areas to help ensure a successful migration that enhances performance, security, and business continuity:

- ▶ Security
 - Security is a top priority in any modernization effort. As applications evolve and integrate with cloud platforms, application programming interfaces (APIs), and open-source technologies, implement robust security measures:
 - Zero trust architecture (ZTA): Adopt a model that assumes no entity (internal or external) is automatically trusted.
 - Data encryption: Encrypt data at rest and in transit by using strong algorithms.
 - Identity and access management (IAM): Use multi-factor authentication (MFA) and role-based access control (RBAC) to prevent unauthorized access.
 - Security audits and compliance: Conduct regular vulnerability assessments and help ensure compliance with industry standards (for example, GDPR, HIPAA, and PCI-DSS).
 - Use of open-source technologies
- Open-source software enhances flexibility, reduces costs, and fosters innovation. However, it requires careful planning:
 - Containers and Kubernetes: Use containerization (such as Docker) and orchestration (such as Kubernetes or Red Hat OpenShift) to improve portability and scalability.
 - Open-source databases: Consider PostgreSQL, MariaDB, and MongoDB as cost-effective alternatives to proprietary databases.
 - Security and maintenance: Apply regular fixes and updates to mitigate vulnerabilities.
 - Support models: Evaluate whether community support is sufficient or if enterprise-grade support (for example, Red Hat or SUSE) is necessary for critical workloads.
- ▶ Storage options
 - Modern applications require scalable and flexible storage solutions to manage growing data volumes:
 - Hybrid and multi-cloud storage: Combine on-premises and cloud storage to optimize scalability and cost.
 - Software-defined storage (SDS): Abstract storage resources to enable automation and efficient resource usage.

- NVMe and Flash Storage: Use high-performance storage to reduce latency and improve responsiveness.
- Backup and Archival Strategies: Implement snapshot-based backups, tiered storage, and immutable backups to protect against ransomware.
- ▶ High availability and disaster recovery (HADR)

Help ensure business continuity through robust HADR strategies:

 - Redundancy and failover: Deploy HA clusters with automated failover to minimize downtime.
 - Disaster recovery planning: Define Recovery Time Objectives (RTOs) and Recovery Point Objectives (RPOs).
 - Cloud-based DR: Use cloud-based DR solutions (for example, IBM Cloud, AWS, or Azure) for flexibility and rapid recovery.
 - Continuous monitoring and testing: Regularly test HADR strategies to help ensure effectiveness during outages.
- ▶ Usage of code assistants

AI-powered code assistants improve development speed, code quality, and team collaboration:

 - Automated code generation and refactoring: Tools like IBM Watsonx Code Assistant®, GitHub Copilot, Tabnine, and OpenAI Codex suggest code snippets, complete functions, and generate code sections to accelerate development and reduce errors.
 - Code review and quality assurance: Automatically detect bugs, inefficiencies, and code smells, helping ensure consistent quality.
 - Intelligent debugging and testing: Identify and fix bugs, generate unit tests, and automate testing to cover edge cases.
 - Simplified maintenance: Help with reviewing earlier code, suggesting refactoring, and integrating modern technologies.
 - Collaboration and knowledge sharing: Promote consistent coding practices and help onboard new developers with context-aware suggestions.

Modernizing applications and infrastructure requires focusing on key areas to improve performance, security, and business continuity. Security efforts should prioritize ZTA, data encryption, and regular audits. Open-source technologies offer flexibility and cost savings but require careful maintenance and support planning. Scalable storage solutions, such as hybrid cloud and SDS, are essential for managing growing data volumes. HADR strategies, including redundancy and cloud-based solutions, help ensure business continuity. AI-powered code assistants enhance development efficiency, automate tasks, and maintain code quality while supporting team collaboration. The following sections of this chapter explore each of these areas in detail.

2.2 Security

IT security is critical in today's digital landscape. As organizations increasingly depend on technology, protecting sensitive data and preventing cyberattacks must remain a top priority. To address modern security challenges, every layer of the IT hardware and software stack must be secured.

IBM Power customers benefit from a highly reliable server platform that enables innovation and faster time to market without compromising security. The IBM Power Architecture and ecosystem deliver advanced security capabilities across multiple layers:

- ▶ **Hardware-level security**

IBM Power servers incorporate advanced, hardware-based security features:

- Tamper-resistant design: Built-in protections prevent unauthorized access and modifications at the processor level.
- Secure cryptographic engines: Enable strong encryption for data at rest and in transit.
- Trusted boot: Helps ensure that only authorized software components load during system startup.

- ▶ **Virtualization-level security**

The IBM Power hypervisor is designed for security and resilience:

- Workload isolation: The hypervisor securely isolates virtual machines (VMs) on a single physical server.
- Secure management: The Hardware Management Console (HMC) provides centralized, secure control of Power servers.

- ▶ **Operating system-level security**

IBM Power supports secure operating systems, including AIX, IBM i, and Linux on Power:

- Authentication and access Control: Built-in features manage user access and permissions.
- Encryption support: Native tools support data protection.
- IBM PowerSC: Provides a comprehensive solution for managing security policies, monitoring threats, and enforcing compliance.

- ▶ **Security management and network protection**

Effective security also requires strong management and network controls:

- Data encryption: Encrypt data both at rest and in transit.
- Network security: Implement firewalls, intrusion detection systems, and other protective measures to safeguard infrastructure.

IBM PowerSC support

Power10 processor-based servers benefit from integrated security management capabilities that are provided by IBM PowerSC, which is a comprehensive software portfolio for managing security and compliance across all IBM Power platforms running AIX, IBM i, or supported Linux distributions.

IBM PowerSC continues to evolve, introducing new features to help organizations manage security end-to-end across the stack and stay ahead of emerging threats. PowerSC 2.0 includes support for Endpoint Detection and Response (EDR), host-based intrusion detection, block listing, and Linux platform support.

By combining hardware-based protections, secure software environments, and centralized management tools, IBM Power servers offer a robust foundation for IT security.

For more information, see *IBM Power Security Catalog*, SG24-8568. This publication outlines best practices, such as conducting regular security audits, keeping operating systems and applications up to date with the latest fixes, and implementing strong user authentication and authorization policies

2.2.1 Encryption

Encryption on IBM Power servers is a critical component of the platform's comprehensive security strategy. IBM Power servers employ a layered approach to encryption, protecting data both at rest and in transit. The platform supports multiple encryption methods, including hardware-based encryption, software-based encryption, and key management services.

Transparent memory encryption

IBM Power solutions feature transparent memory encryption that is designed to deliver end-to-end security that meets the stringent requirements of modern enterprises and supports crypto acceleration, quantum-safe cryptography, and Fully Homomorphic Encryption (FHE). These advanced encryption technologies help ensure that IBM Power servers remain secure today and resilient against emerging threats.

Figure 2-1 shows the implementation of transparent memory encryption.

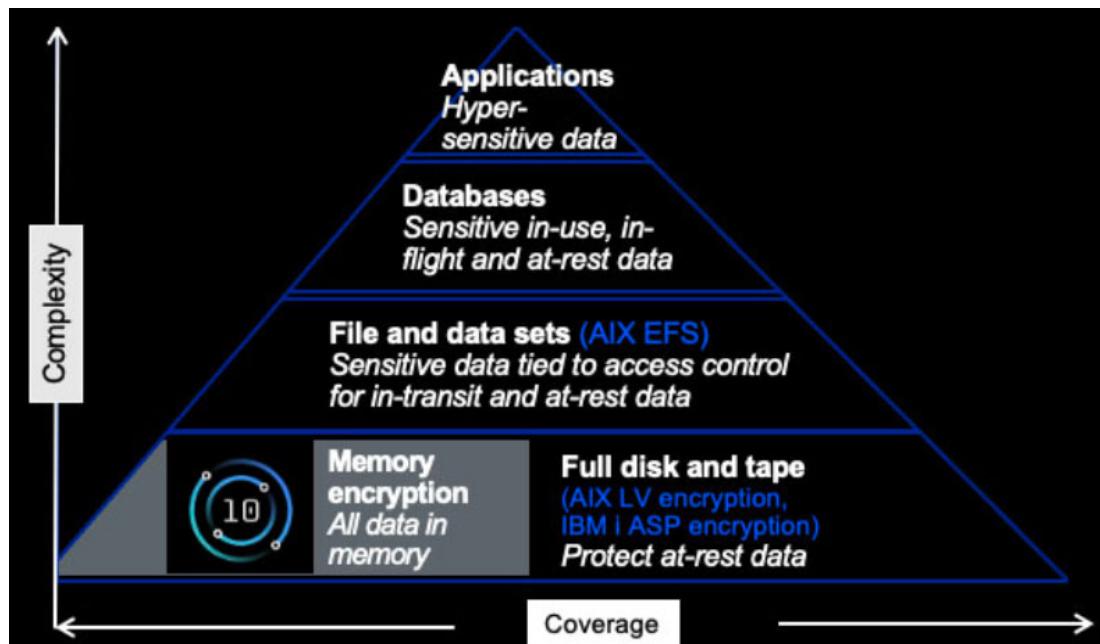


Figure 2-1 Protecting data with memory encryption

Accelerated encryption for the newest IBM Power server model has 2.5x faster Advanced Encryption Standard (AES) crypto performance per core than IBM Power E980 technology. This performance boost enables organizations to benefit from transparent memory encryption without requiring additional management setup.

At the core of this capability is the Power10 MCU, which serves as the interface between the on-chip symmetric multiprocessing (SMP) interconnect fabric and the Open Memory Interface (OMI) links. The Power10 MCU encrypts and decrypts all system memory traffic by using AES-based encryption.

IBM Power10 supports multiple AES encryption modes to address different memory protection needs:

- ▶ **AES XTS Mode**

Xor–Encrypt–Xor Tweakable Block Cipher with Ciphertext Stealing (XTS) provides strong encryption for persistent memory. Persistent DIMM technology retains data even when power is off, making it vulnerable to physical theft. AES XTS helps protect against such threats by encrypting data that is stored in memory modules. This mode is supported for future use when persistent memory solutions become available for IBM Power servers.

- ▶ **AES CTR Mode**

Counter (CTR) mode offers low-latency encryption for volatile memory. Although it is not as strong as XTS, CTR mode is ideal for protecting data in active memory, especially in cloud environments where physical access risks are higher. It helps prevent unauthorized access through memory interfaces. The goal is to protect against physical attacks, which becomes increasingly important in the context of cloud deployments.

2.2.2 Quantum encryption

IBM Power servers are designed to support the cryptographic demands of the quantum era. These systems are optimized to efficiently handle quantum-safe cryptography and FHE, providing a future-ready security foundation.

The software libraries that enable these advanced cryptographic techniques are optimized for the Power10 processor's instruction set architecture (ISA) and are, or will be, available through open-source communities. Future generations of IBM Power servers will continue to expand on this foundation, incorporating more functions to support quantum-safe computing at scale.

Quantum-safe encryption

Quantum-safe encryption (QSE), also known as post-quantum cryptography (PQC), refers to cryptographic methods that are designed to remain secure against both classical and quantum computing threats. As quantum computing advances, existing cryptographic systems might become vulnerable, making QSE essential for protecting sensitive data, communication channels, and digital identities.

The urgency to adopt QSE stems from two key concerns:

- ▶ **Harvest Now, Decrypt Later (HNDL):** Adversaries might intercept and store encrypted data today, with the intent to decrypt it once quantum capabilities become available.
- ▶ **Q-Day:** This anticipated milestone marks the point when quantum computers can break widely used encryption algorithms, rendering many current systems insecure.

Migrating to QSE is a complex, long-term process that might take over a decade due to the intricacies of IT infrastructure and organizational structures. Delaying adoption increases the risk of exposure, especially for financial institutions, critical infrastructure providers, and government agencies. A successful quantum attack might compromise confidential data, disrupt operations, and erode stakeholder trust.

To help ensure long-term cybersecurity resilience, organizations must begin evaluating and implementing QSE solutions now. IBM Power10 supports quantum-safe algorithms, helping organizations prepare for the quantum era with robust, forward-looking security.

Fully Homomorphic Encryption

FHE enables computations to be performed directly on encrypted data without requiring decryption. This capability helps ensure that sensitive information remains confidential throughout the processing lifecycle.

FHE operates at the software level and relies on advanced mathematical algorithms to perform operations on ciphertexts. Implementing FHE requires specialized libraries and frameworks, which are optimized for the IBM POWER processor ISA.

Although FHE offers strong data protection, it is computationally intensive and might impact performance compared to conventional hardware-based encryption methods because of the complexity of the underlying algorithms.

2.3 Open-source software on IBM Power

Open-source software (OSS) is source code that is developed and maintained through open collaboration and community-driven development. OSS is different than proprietary software, which is owned and controlled by a single entity. Anyone can use, examine, alter, and redistribute OSS under a specific license. The source code is available at no charge.

Some key points about OSS:

- ▶ Accessible source code

The core feature is that the software's source code is publicly accessible, allowing anyone to understand how it works and make changes.

- ▶ Community-driven development

Open-source projects often rely on a large community of developers who contribute to improving and enhancing the software.

- ▶ Licensing

OSS is distributed under specific licenses that define how users can modify and distribute the code.

OSS can offer flexibility, cost-effectiveness, transparency, and the ability to customize software to specific needs.

OSS now plays a vital role in computing, with open source technologies providing the foundation of the internet, business computing, and personal computing. OSS is widely used across many industries. Here are some examples of OSS:

- ▶ Linux
- ▶ Mozilla Firefox
- ▶ Apache HTTP Server
- ▶ LibreOffice

Usually, the source code for OSS is available on a code sharing platform such as GitHub (for more information, see 6.4.1, “GitHub” on page 201). There are many mirrors and public servers that provide OSS, such as various universities or companies. One example is [Oregon State University's Open Source Lab](#), which provides OSS for IBM Power, IBM Z, and LinuxONE.

Support for open-source software

When you use OSS for your business applications, understand that many OSS projects are offered “as-is.” The source code is available, but dedicated vendor support is not included. Users encountering issues, bugs, or needing new features are typically expected to consult documentation and use the community-driven development process.

Some OSS projects offer enterprise support, generally for a fee. For example, Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server both provide enterprise support for their distributions. Fedora and openSUSE are also OSS Linux distributions, but they lack formal enterprise support. You can buy a support contract for some open-source projects.

Open source at IBM

IBM works with many open-source communities that drive today’s businesses. IBM developers are collaborators and committers, encouraging open governance, contributing code, helping with licensing, and pushing the technology forward.

IBM demonstrates a [sustained commitment to open source innovation](#) while delivering a broad portfolio of offerings based on open source, and helping to build sustainable, thriving communities and ecosystems around open source projects that matter to IBM clients.

Open source on Power

IBM Power servers use processors that support both big-endian and little-endian operating systems. These systems are supported by AIX, IBM i, and Linux (ppc64le). IBM provides built-in support for OSS on both AIX and IBM i. Because Linux is based on an open-source foundation, OSS is supported by default.

To run OSS on AIX, use the *AIX Toolbox for Open Source Software*, which offers a collection of pre-built open source packages that are designed for AIX. You can install and manage these packages by using tools such as DNF (a package manager) to access a wide range of popular open source applications. For more information, see the [AIX Toolbox overview](#).

Similar to AIX, IBM i supports many open-source products:

- ▶ For more information about how to install open-source packages, see [RPM pile for IBM i releases in standard support](#).
- ▶ As part of the IBM i software maintenance agreement (SWMA), IBM i has an active open-source community. For more information, see the [IBM i Open Source Resources](#) page.
- ▶ For general documentation about open source, see the [IBM i open source documentation](#) and the [IBM i Open Source Resources](#) page.

IBM supports the installation of IBM RPM packages on IBM i. This support includes assistance with software downloads and related tools such as Access Client Solutions. However, it does not include support for the actual usage or defects of these packages.

When evaluating open source applications for IBM Power hardware, review the project’s GitHub page. Carefully read the documentation to determine whether the project supports different CPU architectures, such as ppc64le. It is also helpful to review GitHub Issues and Pull Requests (PRs) to see whether there has been any discussion about supporting IBM Power hardware binary files or packages for IBM Power servers.

One helpful tool to find OSS is the [IBM Open Source POWER availability tool](#).

2.4 Storage options

IBM Power servers support attachment to all types of storage. The type of storage that is used depends on the solution's purpose and objectives.

The following sections describe the various storage types that are commonly found in modern technology environments and provide suggestions for provisioning storage with IBM Power servers.

2.4.1 File storage

File systems organize data in a hierarchical structure of directories and files, making them intuitive for everyday tasks. They were originally designed for direct-attached storage devices that were dedicated to individual servers. Today, file systems also support remotely attached and shared storage through network-attached storage (NAS) devices, which can be provisioned to one or more servers. These devices connect over Ethernet by using protocols such as Network File System (NFS) or Common Internet File System (CIFS). Input/output (I/O) access to the file system occurs through reading/writing individual files from the shared namespace, enabling multiple users to access data simultaneously.

NAS storage solutions are suitable for smaller environments. However, as the number of servers accessing the storage increases, performance can degrade. IBM offers two highly scalable file storage solutions: IBM Storage Scale and IBM Ceph.

IBM Storage Scale, formerly known as General Parallel File System (GPFS), is a high-performance clustered file system that is designed for large-scale environments. It can be deployed in shared-disk, shared-nothing distributed parallel modes, or a combination of both. IBM Storage Scale enables concurrent high-speed file access for applications running on multiple nodes within a cluster. It supports features such as Information Lifecycle Management (ILM), Active File Management (AFM), and Clustered NFS (CNFS).

IBM Storage Scale is used by many of the world's largest commercial enterprises and supercomputers, including the Summit supercomputer at Oak Ridge National Laboratory. For more information, see 2.4.6, “IBM Storage Scale” on page 63.

IBM Ceph is an enterprise-class SDS solution that is built on the open-source Ceph platform. It provides a scalable, reliable, and flexible foundation for managing block, file, and object storage. IBM Ceph is suitable for cloud infrastructure and web-scale object storage. It offers features such as data replication, erasure coding, and self-healing capabilities. Supported interfaces include RESTful APIs (S3/Swift), block device interfaces, and file system interfaces. IBM Ceph is designed to manage vast amounts of data, making it ideal for modern storage requirements. For more information, see *IBM Storage Ceph Concepts and Architecture Guide*, REDP-5721.

For more information on file system solutions, see:

- ▶ [IBM Storage Scale](#)
- ▶ [Introduction to IBM Storage Ceph](#)

2.4.2 Block storage

Block storage, or block-level storage, organizes data into uniform blocks for efficient use in storage area networks (SANs) or cloud platforms. This method divides data, such as files or database entries, into equally sized blocks, optimizing placement on physical storage for rapid access and retrieval. Data interaction occurs through reading/writing individual blocks, allowing the system to distribute data strategically for optimal performance. This approach is suitable for high-transaction environments with large databases that require speed and efficiency.

Block storage typically uses logical volumes (LVs) provisioned to servers within a SAN, employing protocols such as SCSI or Fibre Channel (FC). Modern solutions increasingly use solid-state drives (SSDs) instead of traditional hard disk drives. The NVMe protocol further enhances performance by increasing throughput and reducing latency, making it ideal for high-performance enterprise workloads.

Note: For more information about block storage, see [What is block storage?](#)

2.4.3 Object storage

Object storage systems store data in a flat namespace that can scale to trillions of objects. This architecture is optimized for unstructured data, such as documents, images, audio, and video files. Object storage simplifies data access and supports modern applications by enabling access through various methods, including mobile devices and web applications.

Most cloud storage solutions on the market use an object storage architecture. Notable examples include the following ones:

- ▶ Amazon S3, introduced in March 2006
- ▶ Microsoft Azure Blob Storage
- ▶ Rackspace Cloud Files, whose code was donated to the OpenStack project in 2010 and released as OpenStack Swift
- ▶ Google Cloud Storage, which was released in May 2010
- ▶ OpenStack Swift is an open-source, object storage system that is widely used in cloud environments.

For more information, see [Object storage systems](#), [Object storage](#), and [What is object storage?](#)

2.4.4 Container storage terminology and concepts

Kubernetes provides various volume types to support storage for containers. Each volume type is designed for specific use cases within a containerized environment. A container can use multiple volume types simultaneously.

This section describes the different volume types that are available to containers in Kubernetes.

Ephemeral volumes

During a container's run time, the storage that is used is ephemeral, which means that it is self-contained and portable but not persistent unless explicitly defined by using the container's StorageClass property. An ephemeral read/write layer is created to handle all written data, but this data is not retained. When the container stops (either intentionally or due to failure), the ephemeral data layer is deleted along with the container.

When the container restarts, it is treated as a new instance of the container image, and any data that was written to the previous ephemeral layer is lost.

Kubernetes supports several types of *ephemeral volumes*, each designed for specific use cases:

- ▶ emptyDir: They are empty when they are created at Pod startup. The storage is provided locally from the Kubelet base directory (typically the root disk) or from RAM.
- ▶ configMap, downwardAPI, and secret: Used to inject Kubernetes data into a Pod.
- ▶ image: Allows mounting of container image files or artifacts directly into a Pod.
- ▶ Container Storage Interface (CSI) ephemeral volumes: Provided by CSI drivers that support ephemeral volume features.
- ▶ Generic ephemeral volumes: Supported by any storage driver that also supports persistent volumes (PVs).

The emptyDir, configMap, downwardAPI, and secret volumes are managed by the Kubelet on each node and are considered local ephemeral storage.

For more information about ephemeral storage, see [Ephemeral Volumes](#).

Persistent volume

A *PV* is a low-level representation of a storage volume that is assigned to a container. As the name implies, data that is stored on a PV remains available even after the container is removed and can be accessed when the container is reinitialized. This persistence is achieved through persistent volume claims (PVCs) (see “Persistent volume claims” on page 62, which are defined in the container specification).

PVs are defined by a Red Hat OpenShift PersistentVolume API object. This object represents a piece of existing storage in the cluster, either statically provisioned by a cluster administrator or dynamically provisioned by using a StorageClass object.

PVs function as volume plug-ins, similar to standard volumes, but with a lifecycle that is independent of any individual Pod. A PV object captures the implementation details of the underlying storage, whether it is NFS, iSCSI, or a cloud-provider-specific storage system. A PV can represent any storage that is accessible by the hardware and operating system.

PVs can be provisioned with the following access modes:

- ▶ ReadWriteOnce (RWO): The volume can be mounted as read/write by a single node.
- ▶ ReadOnlyMany (ROX): The volume can be mounted as read-only by multiple nodes.
- ▶ ReadWriteMany (RWX): The volume can be mounted as read/write by multiple nodes.
- ▶ ReadWriteOncePod (RWOP): The volume can be mounted as read/write by a single Pod.

Red Hat OpenShift Container Platform supports the following PV plug-ins:

- ▶ AWS Elastic Block Store (EBS)
- ▶ AWS Elastic File Store (EFS)
- ▶ Azure Disk
- ▶ Azure File

- ▶ Cinder
- ▶ FC
- ▶ Google Cloud Platform (GCP) Persistent Disk
- ▶ GCP Filestore
- ▶ IBM Power Virtual Server Block
- ▶ IBM Cloud VPC Block
- ▶ HostPath
- ▶ iSCSI
- ▶ Local volume
- ▶ NFS
- ▶ OpenStack Manila
- ▶ Red Hat OpenShift Data Foundation
- ▶ CIFS/SMB
- ▶ VMware vSphere

Local volume

Local volumes are PVs that represent locally mounted file systems. They are the lowest level of physical volume attachment for a worker node. They are equivalent to attaching SAN block storage or making Logical Unit Numbers (LUNs) available to a worker node or VM.

In earlier versions of Red Hat OpenShift, administrators had to perform additional manual steps to make these volumes visible and usable. Now, they can use the Red Hat OpenShift Local Volume Operator to manage this process more efficiently.

Persistent volume claims

A *PVC* is specific to a Red Hat OpenShift project and is created by developers to access a PV. PVs are not tied to a single project; they are available across the entire Red Hat OpenShift Container Platform cluster and can be claimed from any project.

After a PV is bound to a PVC, it cannot be bound to more PVCs. This binding effectively scopes the PV to the namespace of the project that made the claim.

A PVC acts as a binding between a pod and a PV. It defines the requested PV type and access mode and is bound to the same namespace as the pod. Kubernetes searches for a PV that matches the criteria that are defined in the PVC. If a suitable PV is available, Kubernetes binds the claim to that PV.

PVCs can request specific sizes and access modes, such as RWO, ROX, or RWX.

Note: For more information about Red Hat OpenShift Storage, see this [partner page](#).

Storage class

A *storage class* enables the dynamic provisioning of PVs when a PVC is made by a Pod or defined in a YAML file. StorageClass abstracts the underlying storage provider.

Each StorageClass includes a back-end provisioner that determines which volume plug-in is used to provision PVs. The dynamic behavior of the storage class is enabled by CSI drivers, which are specific to the storage platform or cloud provider. These drivers allow Kubernetes to access the physical storage.

Each storage back end has its own provisioner, which is defined in StorageClass through the **provisioner** attribute. The autonomy that is provided by CSI improves responsiveness, scalability, and platform management. It also enables more efficient usage of the underlying infrastructure.

2.4.5 IBM Block Storage CSI driver

The CSI is designed as an open specification for exposing block and file storage systems to container orchestration platforms such as Kubernetes and Red Hat OpenShift.

IBM provides the IBM Block Storage CSI driver, which Kubernetes uses for PVs, dynamic provisioning of block storage, and volume snapshots. This driver supports the following storage systems:

- ▶ IBM DS8000® family
- ▶ IBM FlashSystem A9000/R family
- ▶ IBM Spectrum® Virtualize based block storage
- ▶ IBM Storage Scale CSI driver for file-based storage

Note: For more information about the IBM Block Storage CSI driver, see *Using the IBM Block Storage CSI Driver in a Red Hat OpenShift Environment*, REDP-5613.

2.4.6 IBM Storage Scale

IBM Storage Scale (formerly IBM Spectrum Scale) is a high-performance, scalable storage solution that is designed to manage and store large volumes of data across distributed environments. It provides a unified storage platform that supports file, object, and block storage.

IBM Storage Scale is intended for enterprises with demanding workloads, such as AI, analytics, and high-performance computing (HPC). It is IBM's strategic high-performance parallel file system and serves as a shared storage platform for end-to-end enterprise collaboration, data platforms, big data analytics, and AI workflows (see Figure 2-2).

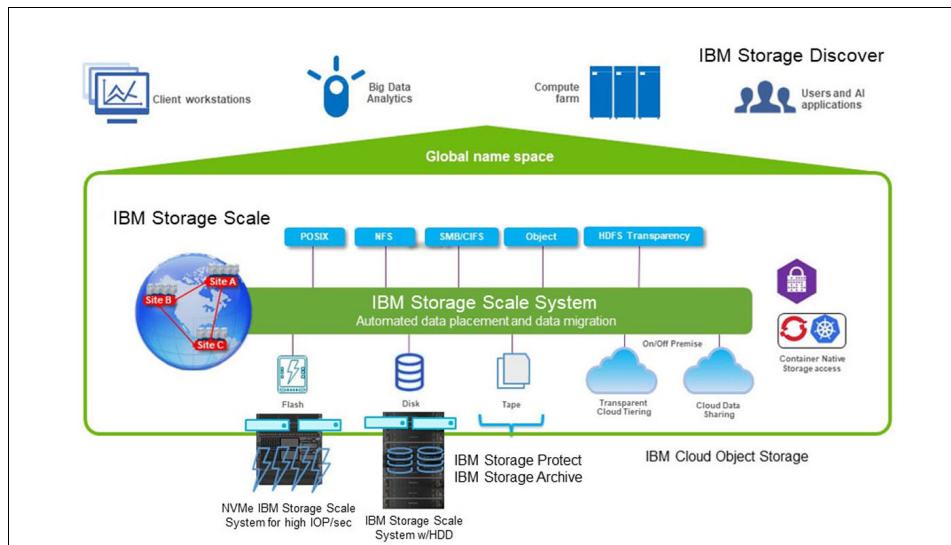


Figure 2-2 IBM Storage Scale System overview

IBM Storage Scale is designed to deliver the following key value propositions:

- ▶ Simplified data management by supporting enterprise workflows on a unified enterprise data platform.
- ▶ A single global namespace that enables enterprise-level data access over high-performance networks.
- ▶ Intelligent automatic tiering of data between storage pools, external tape systems, object-based storage, and cloud resources. This capability supports cost-effective storage by automatically managing and tiering data across different storage classes.

Figure 2-3 illustrates some of the solutions that use IBM Storage Scale.

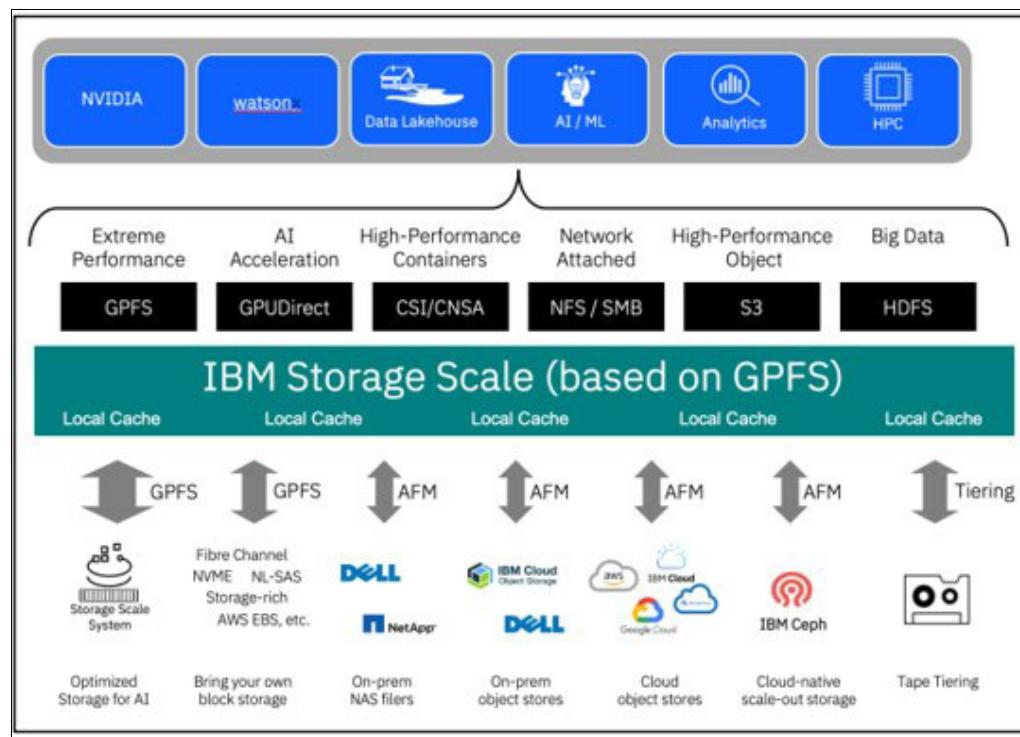


Figure 2-3 IBM Storage Scale solutions

IBM Storage Scale software provides organizations with a global data platform that is optimized for demanding unstructured data workloads. It is based on a massively parallel file system and can be deployed on multiple hardware platforms, including x86, IBM Power, IBM Z mainframes, Application Resource Management (ARM)-based POSIX clients, VMs, and Kubernetes.

IBM Storage Scale is a software-defined file and object storage solution for both structured and unstructured data. The IBM Storage Scale System 6000 is a hardware implementation of the software and is optimized for AI, HPC, analytics, and hybrid cloud workloads.

Figure 2-4 on page 65 shows the IBM Storage Scale System 6000, which can deliver up to 310 GBps throughput.



Figure 2-4 IBM Storage Scale System 6000

The IBM Storage Scale System 3500 is designed for customers who require an enterprise-ready entry-level, or mid-level system.

The IBM Storage Scale client achieves high performance by performing real-time, parallel I/O to all IBM Storage Scale data servers, storage volumes, and NSDs simultaneously. An IBM Storage Scale cluster can scale by adding nodes, whether they are IBM Storage Scale clients or IBM Storage Scale data servers.

Users are unaware of the physical distribution of data within the IBM Storage Scale data server storage pools. The IBM Storage Scale policy engine automatically determines data placement at the time of import and can transparently move data between storage pools while the data remains active and accessible.

The IBM Storage Scale parallel file system enables enterprises to manage large volumes of data while continuously balancing workloads and storage. It distributes I/O operations and data evenly within or across storage pools.

The preferred method for accessing IBM Storage Scale data is to install the IBM Storage Scale client on each workstation or server that requires access. The client enables high-performance parallel throughput by using multiple threads and communicating with multiple data servers. IBM Storage Scale also helps ensure full read/write data integrity for multiple users accessing the file system concurrently.

IBM Storage Scale for NVIDIA

IBM Storage Scale is the preferred storage system for NVIDIA solutions. The IBM Storage Scale System is an NVIDIA-certified, ultra-performance solution that supports AI innovation and scales seamlessly from NVIDIA DGX BasePOD to the largest DGX SuperPOD installations.

Deployed by thousands of organizations for GPU acceleration and AI workloads, IBM Storage Scale System delivers six nines (99.9999%) of data reliability, cyber resiliency, and multi-protocol data pipelines for demanding enterprise environments. The software-defined IBM Storage Scale integrates and tiers data, enabling organizations to use a global data platform that transforms data-intensive AI workloads into actionable insights, as illustrated in Figure 2-5.

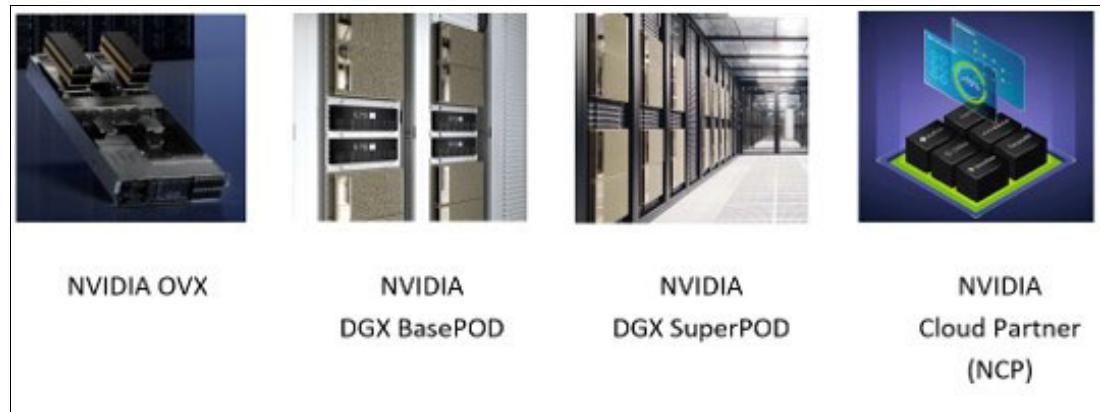


Figure 2-5 NVIDIA systems solutions with IBM Storage Scale

Unstructured and semi-structured data from AI workloads, advanced analytics, data lakes, and other data-intensive applications must be stored in distributed file and object systems to help ensure accessibility across geographically dispersed applications, services, and devices.

IBM Storage Scale software is designed to meet these requirements through global data abstraction services. These services provide connectivity across multiple data sources and locations, enabling organizations to unify data regardless of where it is, including in third-party storage environments.

2.4.7 IBM Storage Fusion

IBM Storage Fusion is a container-native, hybrid cloud data platform that simplifies deployment and data management for Kubernetes applications on Red Hat OpenShift Container Platform. It is designed to meet the storage requirements of modern, stateful Kubernetes applications and to streamline the deployment and management of container-native applications and their data.

IBM Storage Fusion is an advanced storage and backup solution that enhances data accessibility and availability across hybrid cloud environments. It enables organizations to expand data availability across complex hybrid clouds, improving business performance and resilience. With IBM Storage Fusion, organizations manage a single copy of data, eliminating the need to duplicate data when moving application workloads across the enterprise. This simplification supports streamlined analytics and AI operations.

Before installing any IBM Cloud Pak product, a Red Hat OpenShift cluster administrator must properly configure storage and understand any limitations that are associated with the selected storage. Not all services support all types of storage. Although it is possible to configure a mix of storage providers, using a single provider simplifies administration.

IBM Cloud Pak products require high-performance storage solutions and use CSI drivers from platform providers such as IBM Storage Scale, AWS EFS, and Azure Files. Caution is advised when using open-source storage providers because they might have limitations, such as connection limits.

IBM Storage Fusion includes high-performance Storage Scale SDS management software and consolidates all IBM Storage Protect solutions, providing a comprehensive solution for storage and data protection across all container workloads, as shown in Figure 2-6.

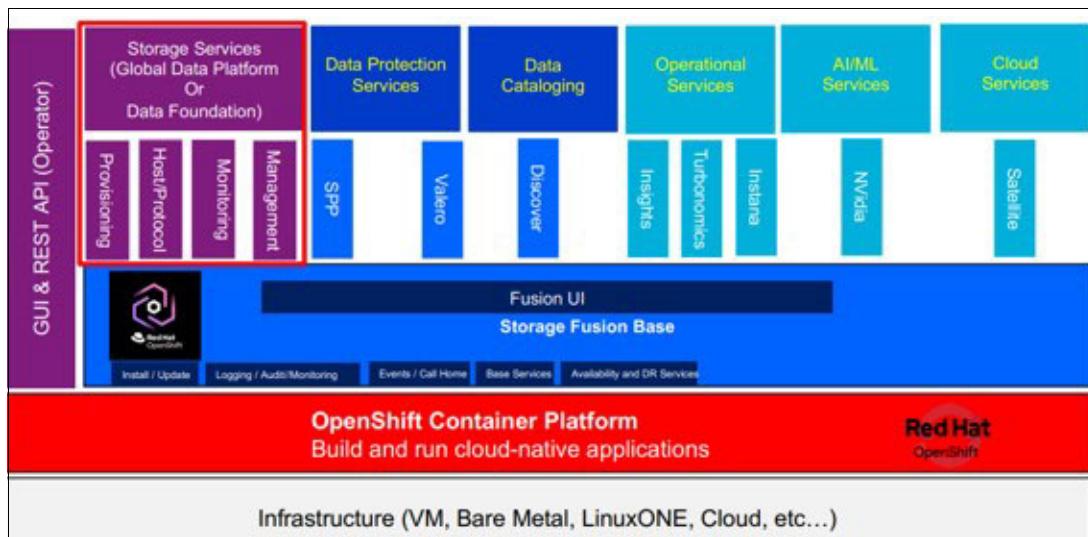


Figure 2-6 IBM Storage Fusion general architecture

IBM Fusion offerings

There are two offerings for IBM Storage Fusion:

- ▶ IBM Fusion
- ▶ IBM Fusion HCI System

IBM Fusion

IBM Fusion is an SDS management software with protection, backup, and caching elements that can run on existing hardware resources. IBM Fusion is supported on several platforms and clouds.

IBM Fusion HCI System

IBM Fusion HCI is a purpose-built, hyper-converged infrastructure that is designed to deploy bare-metal Red Hat OpenShift container management and deployment software alongside IBM Fusion software. To support consistent and rapid deployment, it features an appliance form factor and integrates SDS to meet the needs of modern, stateful Kubernetes applications.

Built on a storage platform that includes essential components for mission-critical containers and hybrid cloud environments, IBM Fusion HCI delivers a comprehensive infrastructure. It provides compute, networking, and storage resources, along with a unified data platform and global data services for Red Hat OpenShift.

Note: To learn more about the IBM Fusion HCI System, see the [IBM Fusion documentation](#).

The storage component of IBM Fusion includes two distinct offerings:

- ▶ Global Data Platform – Containerized IBM Storage Scale

This option provides the following features:

- Storage efficiency with Storage Scale RAID
- AFM for geographic distance extension
- A scale-out parallel file system

Potential use cases include the following ones:

- IBM WatsonX
- Db2 Warehouse
- Metro or regional DR

- ▶ Data Foundation – Ceph with Containerized Services

This option provides the following features:

- Block, file, and object storage (CephRBD, CephFS, or RGW)
- A consistent architecture and storage classes across multiple infrastructures
- Simplified lifecycle management

Potential use cases include basic storage provisioning requirements.

Storage and performance validation

To determine whether the storage that is attached to your Kubernetes or Red Hat OpenShift cluster is compatible with IBM Cloud Pak products and capable of delivering the required I/O performance, you can run validation tools. IBM provides both a storage validation tool and a performance validation tool to assist with this process.

Storage validation tool

Run the IBM Software Hub storage validation tool on your Red Hat OpenShift cluster to verify that your storage configuration is suitable for use with IBM Software Hub. Before installing any IBM Cloud Pak products, set up your environment and run the validation tool to confirm compatibility and readiness.

For more information about the tool, see this [storage validation tool](#).

Performance validation tool

Run the storage performance tool to help ensure that your storage performs as expected and meets the performance guidelines. Use this tool to collect storage performance metrics on your Red Hat OpenShift cluster.

For more information about the tool, see this [k8s-storage-perf GitHub repository](#).

New in IBM Storage Fusion 2.9

IBM Fusion 2.9.0 introduces new features in the following areas:

- ▶ New platform support

IBM Fusion can now be deployed on the AWS ROSA HCP platform.

- ▶ Parallel upgrades

Parallel upgrades are supported for IBM Fusion operator components that are not part of rolling updates. This enhancement allows multiple components to be upgraded simultaneously, reducing downtime and improving system efficiency.

- ▶ Automatic Backup & Restore service upgrade
The Backup & Restore service is now automatically upgraded based on service availability.
- ▶ Multi-cluster enhancements
Improved support for Multi-cluster IBM Fusion by using HCP. Enhanced ability to deploy IBM Fusion services through Red Hat Advanced Cluster Management for Kubernetes.
- ▶ Simplified image mirroring
Image mirroring during installation and upgrade is simplified.
- ▶ Change block detection support
Backup & Restore now supports change block detection for Ceph RBD block volumes. This enhancement allows the DataMover to identify changes without scanning the entire volume, reducing backup times for applications for using Ceph RBD block mode volumes, including Red Hat OpenShift Virtualization VMs.
- ▶ Self-service Backup & Restore
Application users without cluster or IBM Fusion administrative rights can now protect their namespace applications by using IBM Fusion Backup & Restore.

For more information about IBM Storage Fusion, see the [product documentation](#).

2.4.8 IBM Fusion Deployment Options

IBM Fusion is an SDS solution that runs on Red Hat OpenShift. The IBM Fusion HCI product provides a flexible hardware platform that can be integrated into your infrastructure and deployed within hours.

The IBM Fusion SDS product supports environments running on Red Hat OpenShift, either on premises or in cloud environments.

Figure 2-7 illustrates the deployment environments for IBM Fusion.

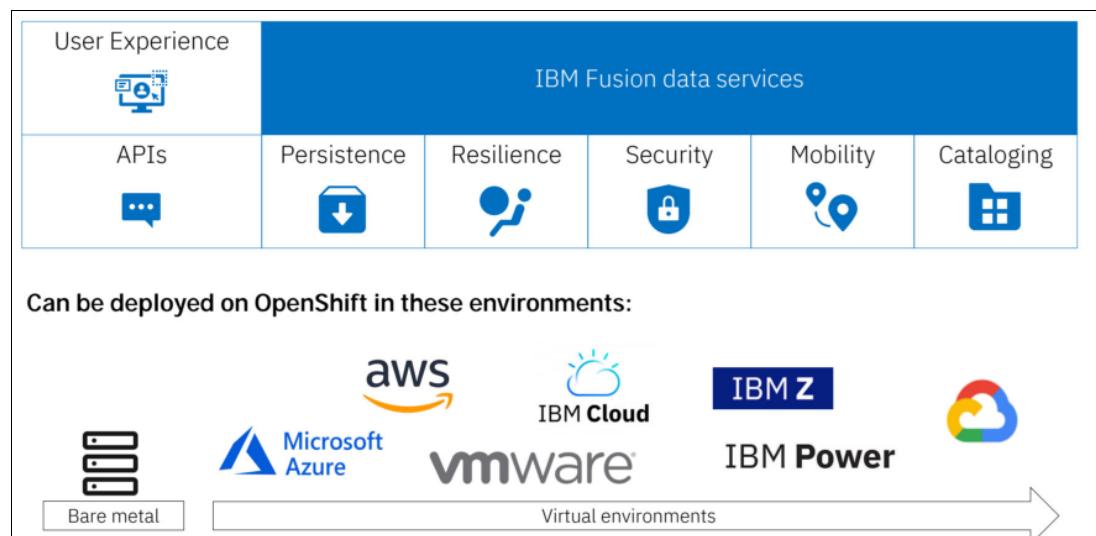


Figure 2-7 IBM Fusion deployment options

For a full list of supported services by platform, see this [support matrix](#).

The next section focuses on IBM Fusion running on IBM Power.

IBM Fusion on Premises IBM Power

IBM Fusion on IBM Power provides all the capabilities of IBM Fusion, as shown in Figure 2-8.

IBM Power Systems					
Platform	Data Foundation	Data Foundation (Discover an existing installation)	Global Data Platform	Backup & Restore	Data Cataloging
IBM Power Systems	Yes	Yes	Yes	Yes	Yes

Figure 2-8 IBM Fusion support for IBM Power

Installation considerations

The following considerations must be addressed before installing IBM Fusion on IBM Power:

- ▶ A Red Hat OpenShift cluster must be installed before you install IBM Fusion. IBM Fusion is available as an operator in the Red Hat OpenShift Container Platform web management console.
- ▶ Help ensure that three additional worker nodes are available. These nodes must meet the minimum CPU, memory, and SAN storage requirements for the IBM Fusion discovery process to detect the storage workers. For system requirements, see [IBM Fusion System Requirements](#). This resource outlines the CPU, memory, and storage requirements based on the IBM Fusion components being configured.
- ▶ Select and configure the worker nodes by using the IBM Storage Fusion GUI.
- ▶ For optimal performance and reliability, use SSD or NVMe devices. HDDs are supported only for development or test environments. For more information about configuring storage, see [IBM Fusion Storage Configuration](#).
- ▶ When a new Red Hat OpenShift cluster is installed on IBM Power, several no-charge operators are available for manual storage provisioning. These operators include CSI drivers for IBM Block Storage and IBM Spectrum Scale, as shown in Figure 2-9.

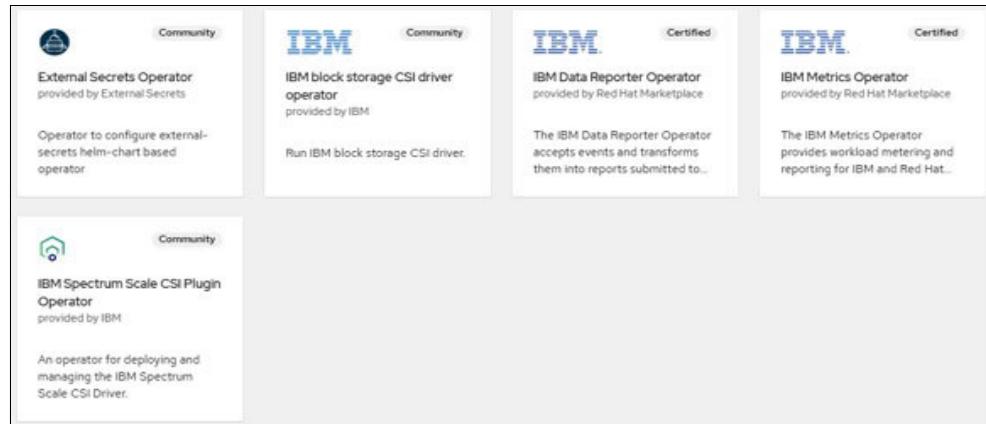


Figure 2-9 Operator catalog default view

When the IBM Operator catalog is enabled, the full suite of IBM products (including IBM Fusion) is available for installation, as shown in Figure 2-10 on page 71.

The screenshot shows the 'OperatorHub' section of the IBM OperatorHub interface. On the left, there's a sidebar with a tree view of categories like AI/Machine Learning, Application Runtime, Automation, Big Data, Business Automation, Cloud Provider, Database, Developer Tools, Integration & Delivery, Logging & Tracing, Modernization & Migration, Monitoring, Networking, OpenShift Optional, Other, Security, Storage, Streaming & Messaging, and Source. Below the tree view is a search bar and a link to 'All Items'. The main area displays a grid of 65 items, each with a thumbnail, name, provider, and a brief description. Some examples include 'canvasbase Modeler Services provided by ibm', 'Common Core Services provided by IBM', 'Data Product Hub Services provided by ibm', 'Decision Optimization for Cloud Pak for Data provided by IBM', 'External Secrets Operator provided by External Secrets', 'IBM AIOps Edge provided by IBM', 'IBM Analytics Engine Powered by Apache Spark Service provided by cpd', 'IBM API Connect provided by IBM', and 'IBM App Connect provided by IBM'.

Figure 2-10 View of IBM operators

- The IBM Fusion operator does not create a storage cluster unless a valid IBM Fusion license is attached to the administrator's IBM Container Registry key. You can provide your own license key or obtain a trial IBM Fusion license from [this location](#).

After obtaining the license key, configure the IBM Fusion pull secret in Red Hat OpenShift. Once the pull secret is configured, you can install IBM Fusion from the operator. For instructions on creating the pull secret, see the relevant IBM Documentation found [here](#).

After installation, you can choose to create a Data Foundation deployment for the ODF Ceph cluster or select the Global Data Platform, which uses IBM Storage Scale, as shown in Figure 2-11.

The screenshot shows the 'Services' section of the IBM Fusion interface. The left sidebar has links for Dashboard, Events, Applications, Storage, Services (which is currently selected), and Settings. The main area is titled 'Services' with a sub-section 'Available'. It shows two service options: 'Data Foundation' and 'Global Data Platform'. 'Data Foundation' is described as 'Provides a foundational data layer for applications to function and interact with data in a simplified, consistent and scalable manner.' 'Global Data Platform' is described as 'This service provides crash consistent backup, access to data from external storage sources and Container Storage Interfaces (CSI) provisioning of high performance storage.' Both services are listed under the 'IBM + Storage' category.

Figure 2-11 Storage configuration choice in IBM Fusion

For the full installation documentation, see [Installing IBM Fusion on premises on IBM Power](#).

2.5 High availability and disaster recovery

In an era where digital technologies support every aspect of life, from critical business operations to personal communication, uninterrupted access to data, applications, and services is essential. The modern world demands a seamless, always-on experience, and any disruption, whether caused by hardware failure, software issues, or unforeseen disasters, can result in frustration and loss.

In this context, HADR emerges as a key component of resilience and business continuity. As modern business demands increase, critical applications must remain continuously available, and systems must be fault-tolerant. However, implementing fault-tolerant systems can be costly. Therefore, organizations require solutions that deliver high availability and disaster recovery (HADR) capabilities in a cost-effective manner.

An HA solution helps ensure that the failure of a single component does not affect application or data availability. This goal is achieved by eliminating single points of failure and minimizing both planned and unplanned downtime.

When HA measures are insufficient, disaster recovery (DR) provides a secondary layer of protection. DR is typically used in response to large-scale failures, such as complete site outages or data corruption caused by cyberattacks. DR processes often involve manual intervention due to the complexity and criticality of the situation. These processes might include restoring from backups, running site failovers, or performing detailed database log analysis to maintain logical consistency and determine whether specific operations should be applied.

Figure 2-12 illustrates failover capability, where one server automatically assumes operations if another server fails.

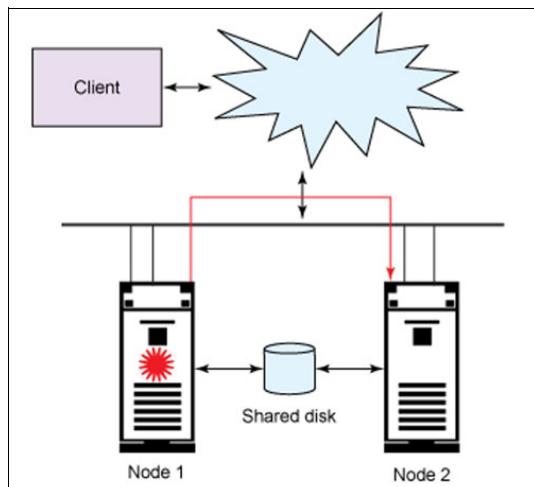


Figure 2-12 Failover capacity for high availability

The configuration that is shown in Figure 2-12 provides a HA solution within a single site. However, if there is a site-level outage, such as one caused by a power failure, fire, flood, or other natural disaster, additional measures are required to recover data and maintain application operations.

Figure 2-13 illustrates a HADR solution that uses a three-site architecture. This configuration includes two on-premises sites that are managed by the client and a third failover site that is hosted in the IBM Cloud, which contains servers and storage systems.

If Site 1 and Site 2 are within the same data center or geographic region, applications can be restarted at Site 2 with minimal or no downtime if there is a planned or unplanned outage at Site 1. If both Site 1 and Site 2 are affected, application operations can be resumed at the remotely located Site 3.

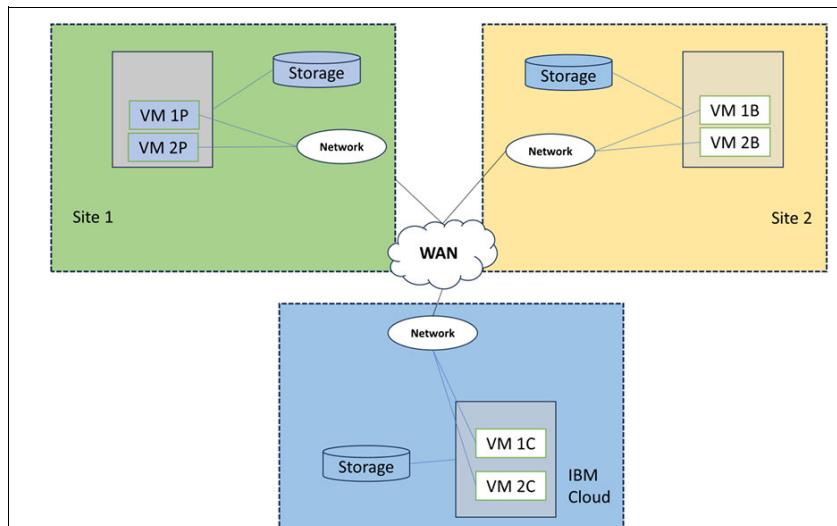


Figure 2-13 Full HADR solution that uses three sites with storage replication

2.5.1 IBM solutions for HADR

IBM Power, including IBM Power servers and IBM Power Virtual Server in the cloud, offers a comprehensive portfolio of HADR solutions. Figure 2-14 provides an overview of the HADR capabilities that are available on IBM Power servers.

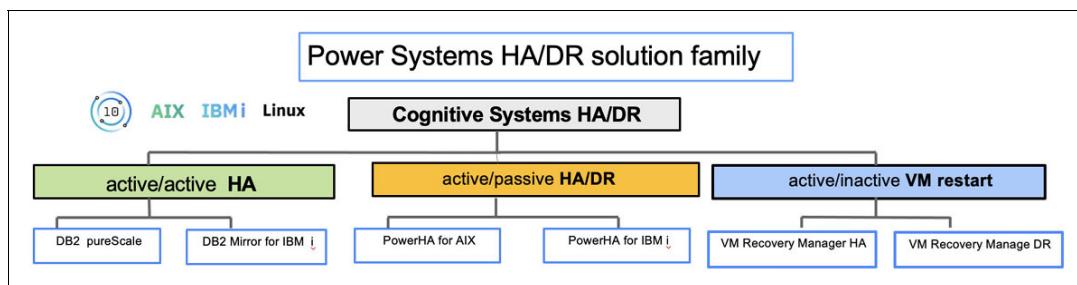


Figure 2-14 IBM Power virtual server HADR solution family

- ▶ IBM PowerHA SystemMirror®

The PowerHA SystemMirror family of solutions is optimized for mission-critical applications that require zero or near-zero annual downtime for both planned and unplanned outages. The PowerHA SystemMirror product line addresses all types of outages, including both software and hardware failures. Each node in the cluster runs at least one active operating system, which allows software updates to be performed on a non-production node.

PowerHA SystemMirror supports both single data center and multi-site configurations. To minimize total outage time for planned and unplanned events, this solution is recommended.

For more information, see 2.5.2, “IBM PowerHA SystemMirror” on page 75.

- ▶ IBM Virtual Machine Recovery Manager (VMRM)

VMRM solutions are best understood in the context of Live Partition Mobility (LPM). A set of logical partitions (LPARs) is virtualized by using IBM PowerVM and Virtual I/O Server (VIOS), enabling partitions to be moved during firmware or hardware maintenance events by using LPM.

If a VM fails, it can be restarted on another server within the cluster. For DR operations, these VMs are replicated by using SAN storage at a secondary location.

For more information, see *IBM Virtual Machine Recovery Manager for IBM Power Cookbook*, SG24-8539.

- ▶ Active-active solutions

IBM offers active-active solutions such as IBM Db2 pureScale® (for AIX) and IBM Db2 Mirror for i. Although both are classified as active-active, they use different approaches:

- Db2 pureScale provides an active-active configuration by using a shared Db2 cluster with distributed lock management, allowing multiple application servers to access the shared database simultaneously.
- IBM Db2 Mirror for i delivers continuous availability for mission-critical applications by synchronously mirroring database updates between two nodes by using Remote Direct Memory Access (RDMA) over Converged Ethernet (RoCE).

Active-active configurations are also available for other database platforms. For example, Oracle Real Application Clusters (RAC) and Oracle Data Guard can be used to build highly available Oracle Database solutions.

Table 2-1 on page 75 highlights the differences between IBM Power HADR solutions.

Table 2-1 HA topology classification.

Technology	Active-active clustering	Active-passive clustering	Active-inactive clustering
Definition	Application clustering: Applications in the cluster have simultaneous access to production data, which eliminates the need for application restarts in a node outage. Certain configurations also support read-only access from secondary nodes.	OS clustering: In this configuration, one OS instance in the cluster has access to the production data. Multiple active OS instances run across all nodes in the cluster. If the production node experiences an outage, the application is restarted on a secondary node.	In this configuration, one VM in a cluster pair has access to the production data. The setup includes a single, logical OS instance and two physical copies. If the primary node experiences an outage, the OS and applications must be restarted on a secondary node. LPM enables the VM to be moved non-disruptively during planned maintenance events.
Outage types	<ul style="list-style-type: none"> ▶ Software, hardware, HA, planned, and unplanned ▶ RTO 0 with limited distance 	<ul style="list-style-type: none"> ▶ Software, hardware, HADR, planned, and unplanned ▶ RTO>0 multi-site 	<ul style="list-style-type: none"> ▶ Hardware, HADR, planned, and unplanned ▶ RTO>0 multi-site
OS integration	Inside the OS	Inside the OS	OS-neutral
RPO	Sync mode only	Sync and async	Sync and async
RTO	Zero	Fast (minutes)	Fast Enough (VM Reboot)
Licensing	N+N ^a	N+1 licensing	N+0 licensing
IBM solution	Db2 pureScale and Db2 Mirror ^b	PowerHA, Red Hat HA, and Linux HA	VMRM HADR

a. N: The number of licensed processor cores on each system in the cluster.

b. Other database vendors have similar active-active solutions.

2.5.2 IBM PowerHA SystemMirror

IBM PowerHA technology enables the deployment of a HA solution that integrates both storage and availability requirements into a single, streamlined configuration with a simple interface. IBM Power continues to focus on delivering solutions that enhance the resilience of IT environments.

IBM PowerHA is available in two editions, Standard and Enterprise, for both IBM AIX and IBM i. Beginning with IBM i 7.5, PowerHA for IBM i is offered as a unified product that includes the features of both the Standard and Enterprise editions.

IBM PowerHA SystemMirror for AIX

IBM PowerHA SystemMirror for AIX provides a cost-effective, commercial computing solution that is designed to enable rapid recovery of mission-critical applications if there are hardware or software failures.

With PowerHA SystemMirror, essential resources remain accessible. For example, a PowerHA cluster can host a database server application that serves client requests by retrieving data from a shared external disk.

This HA solution combines custom software with industry-standard hardware to minimize downtime by quickly restoring services after a failure in the system, a component, or an application. Although recovery is not instantaneous, services are typically restored within 30 - 300 seconds.

In a PowerHA SystemMirror cluster, applications are managed directly by the software to help ensure continuous availability. If a component within the cluster fails, PowerHA SystemMirror automatically transfers the affected application and its associated resources to another node, maintaining uninterrupted service for client processes.

PowerHA SystemMirror provides the following capabilities:

- ▶ Planning and best practices: Includes guidance and documentation for installing and maintaining a HA PowerHA SystemMirror cluster.
- ▶ Automated monitoring and recovery: Continuously monitors all resources on which the application depends and automatically recovers them as needed.
- ▶ Cluster maintenance tools: Offers a comprehensive set of tools for maintaining the cluster while keeping applications available to clients.
- ▶ Simplified cluster setup: Enables quick setup of a basic two-node cluster by using the typical initial cluster configuration SMIT path, or Smart Assists (application configuration assistants).
- ▶ Cluster testing: Includes the Cluster Test Tool to simulate failure scenarios, such as node or network inaccessibility, and evaluate cluster behavior.
- ▶ HA assurance: Eliminates single points of failure to enable the HA features of the AIX operating system.
- ▶ Failure management: Allows administrators to define how the cluster responds to component failures.
- ▶ Secure communications: Helps ensure secure communication between cluster components.
- ▶ Monitoring and diagnostics: Provides tools to monitor PowerHA SystemMirror components and diagnose potential issues.

For more information, see [PowerHA SystemMirror for AIX](#).

IBM PowerHA SystemMirror for IBM i

IBM PowerHA SystemMirror for i delivers comprehensive, integrated clustering solutions for both HADR. As a built-in extension of the IBM i operating system, PowerHA provides robust resiliency for the environment, applications, and data, which help ensure continued access and storage management during both planned and unplanned outages.

With its automation capabilities, intuitive management interface, and seamless integration with IBM i, PowerHA supports HADR across a wide range of IBM i-based systems and storage configurations:

- ▶ Geographic Mirroring
A system-based solution that supports synchronous or asynchronous data transmission. This replication method is independent of specific storage technologies.
- ▶ Metro Mirroring
A synchronous copy services function that replicates independent auxiliary storage pools (IASPs) between cluster nodes by using supported external storage units.
- ▶ Global Mirroring
An asynchronous copy services function that replicates IASPs between cluster nodes by using supported external storage units.
- ▶ Switched logical units
LUN switching enables a supported external storage unit to be moved between systems in a cluster, allowing an IASP to be transferred from one node to another.
- ▶ IBM HyperSwap® and PowerHA
HyperSwap provides a full system solution that mirrors logical units between two IBM System Storage DS8000 systems. PowerHA offers an interface to manage both Full System HyperSwap and IASP-level HyperSwap configurations.
- ▶ IBM FlashCopy® and PowerHA
FlashCopy is a function of IBM System Storage that creates fast, point-in-time copies of data. These copies can be stored, used, or brought online on a separate partition or system. FlashCopy can be combined with other data resiliency technologies and managed through PowerHA.

These solutions can be deployed individually or in combination, depending on the specific requirements of your organization.

For more information about IBM PowerHA System Mirror for i, see the [PowerHA Wiki](#).

2.5.3 IBM Virtual Machine Recovery Manager

VMRM is an automated solution that enables recovery of LPARs by using restart technology. VMRM relies on an out-of-band monitoring and management component that restarts VMs on another server when the host infrastructure fails. Multiple deployment options are available to meet different availability and recovery requirements.

Depending on your needs, VMRM can provide:

- ▶ VMRM HA: An HA solution within a single site or across two sites within metro distances.
- ▶ VMRM DR: A DR solution for managing workloads between a primary (production) site and a secondary (backup or DR) site.
- ▶ VMRM HADR and VMRM HADRHA: Extended options that combine HA and DR capabilities to provide integrated HADR solutions, offering greater flexibility in workload recovery locations.

VMRM differs from traditional clustering technologies, which rely on redundant hardware and software components to enable near real-time failover. VMRM HA is ideal for helping ensure availability across many VMs and, if it meets your RPO requirements, offers a simpler alternative to cluster-based environments due to its reduced complexity.

VMRM HA is based on Simplified Remote Restart (SRR) technology, making it operating system-independent and compatible with IBM AIX, IBM i, and Linux. This simplified management capability extends to DR scenarios through VMRM DR. The HADR and HADRHA options build on VMRM HA and DR to deliver comprehensive, integrated HADR functions.

VMRM HA

HA management is a critical component of business continuity planning. Any downtime in the software stack can lead to revenue loss and service disruption. VMRM HA for Power provides a simple, automated solution for recovering VMs, also known as LPARs.

VMRM HA implements recovery by using SRR. This technology relies on an out-of-band monitoring and management component that restarts VMs on another server when the host infrastructure fails.

Unlike traditional cluster-based technologies that use redundant hardware and software components for near real-time failover, VMRM HA offers a simpler alternative. It is ideal for helping ensure HA across many VMs and is simpler to manage because it avoids the complexities that are associated with clustering environments.

VMRM DR

DR of applications and services is a critical component of helping ensure continuous business operations. VMRM DR for Power provides a simple-to-deploy, automated solution for recovering workloads at the production site.

The VMRM DR solution is based on the concept of IBM Geographically Dispersed Parallel Sysplex® (IBM GDPS®), which optimizes resource usage. Unlike traditional DR solutions, VMRM DR does not require the deployment of backup VMs, helping reduce software licensing and administrative costs.

VMRM DR uses SRR across two sites. This HADR solution relies on an out-of-band monitoring and management component that restarts VMs on alternative hardware when the primary host infrastructure fails.

VMRM HADR

VMRM HADR extends the capabilities of the VMRM DR solution by adding HA features. This solution offers greater flexibility in determining where applications can run after a failure, supporting recovery at both primary and secondary sites.

Key HA features of the VMRM HADR deployment include:

- ▶ LPM support within a site
- ▶ Support for VM and host failures within a site
- ▶ Application failover within a site
- ▶ FC adapter failure handling within a site
- ▶ Network adapter failure handling within a site
- ▶ Monitoring of CPU and memory usage within a site
- ▶ Move operations and failover rehearsal across and within sites
- ▶ Disaster recovery support from one site to another site

VMRM HADRHA

VMRM HADRHA extends the capabilities of both the VMRM DR and VMRM HADR solutions by adding enhanced HA features. This solution enables application recovery within the same site at both the primary and secondary locations, offering greater flexibility and resilience.

Key HA features of the VMRM HADRHA deployment include:

- ▶ LPM support within a site
- ▶ Support for VM and host failures within a site
- ▶ Application failover within a site
- ▶ FC adapter failure handling within a site
- ▶ Network adapter failure handling within a site
- ▶ Monitoring of CPU and memory usage within a site
- ▶ Move operations and failover rehearsal across and within sites
- ▶ DR support from one site to another site

2.5.4 Linux high availability clustering

VMRM uses SRR, making it operating system-neutral and capable of delivering HADR for Linux VMs.

Previously, IBM offered a version of IBM PowerHA Recovery Manager that supported Linux. However, this product is discontinued and no longer supported. The discontinuation coincided with the rise of Pacemaker, an open-source, full-featured Linux HA clustering solution. Pacemaker, with Corosync, which manages cluster node communication and membership, is widely adopted across multiple Linux distributions to deliver robust cluster management capabilities.

Pacemaker

Pacemaker is an open-source HA cluster resource manager that is designed to manage and maintain the integrity of services running on a group of hosts, which is known as a cluster. It detects and recovers from both host- and application-level failures, helping ensure minimal downtime and preserving data integrity.

Pacemaker supports various redundancy configurations, including active-passive and N+1 setups, and can manage nearly any application that can be controlled through scripting. Its ability to handle complex service dependencies, such as ordering and colocation, makes it a versatile solution for maintaining HA in Linux environments.

The Pacemaker architecture is built around the concept of resources, which represent the services that must remain HA. Resource agents (scripts or OS components) start, stop, and monitor these resources based on defined parameters.

Pacemaker also includes fencing capabilities, which are known as Shoot The Other Node In The Head (STONITH), which isolate faulty nodes to prevent data corruption. This goal is typically achieved through intelligent power or network switches that cut power or network access to the affected node.

Pacemaker is widely supported across major Linux distributions, including RHEL and SUSE Linux Enterprise Server. Both distributions integrate Pacemaker into their HA offerings and provide robust tools and documentation for configuring and managing clusters. Red Hat and SUSE offer enterprise-grade support and deployment guidance to help ensure the availability of critical applications.¹

The collaborative development of Pacemaker by the ClusterLabs community (with contributions from Red Hat, SUSE, and others) has led to its widespread adoption in enterprise environments. This broad support underscores reliability and effectiveness of Pacemaker as a HA cluster manager for Linux systems.

¹ Source: <https://www.redhat.com/en/blog/rhel-pacemaker-cluster>

2.5.5 Additional information on HADR solutions

For more information about HADR solutions in IBM Power environments, see the following IBM Redbooks publications:

- ▶ *IBM PowerHA SystemMirror and IBM VM Recovery Manager Solutions Updates*, REDP-5694
- ▶ *IBM PowerHA SystemMirror for AIX Cookbook*, SG24-7739
- ▶ *Using Pacemaker to Create Highly Available Linux Solutions on IBM Power*, SG24-8557

2.6 AI code assistants

Software development is undergoing a profound transformation, which is driven by rapid advancements in AI. AI-powered coding assistants are at the forefront of this shift, enhancing productivity, improving code quality, and streamlining development workflows. These intelligent tools augment human capabilities, redefine best practices, accelerate development cycles, and drive greater efficiency.

According to the Gartner Magic Quadrant for AI Code Assistants,² 90% of enterprise software engineers will use AI code assistants by 2028, which is up from less than 14% in early 2024. Also, Markets and Markets projects that the global market for AI code tools will grow from USD 4.3 billion in 2023 to USD 12.6 billion by 2028, at a compound annual growth rate (CAGR) of 24.0%.³

AI is poised to influence every stage of the software development lifecycle. From code generation and unit test automation to code transformation, bug detection, and autonomous fixes, AI-driven assistants and intelligent agents are revolutionizing how developers work. These technologies enhance efficiency and establish new industry standards, making software development faster, more reliable, and increasingly automated.

The rapid pace of AI innovation has propelled coding assistants from emerging technologies to essential tools in the developer's toolkit. AI-powered assistants, such as IBM Watsonx Code Assistant, provide intelligent code suggestions, automated bug resolution, and code optimization, empowering developers to write clean, efficient code at an accelerated pace.

2.6.1 IBM Watsonx Code Assistant

IBM Watsonx Code Assistant is designed to simplify and automate both new software development and application modernization. It enhances productivity and empowers development teams of all skill levels to support business innovation.

The solution accelerates workflows across multiple programming languages, including Python, Java, C, C++, Go, JavaScript, and TypeScript, by providing generative AI assistance directly within the integrated development environment (IDE). It also supports end-to-end modernization of Java applications and runtime environments, helping improve business agility and reduce technical debt through a customized, automated experience.

² Source:

https://www.gartner.com/doc/reprints?id=1-2IK04MP6&ct=240819&st=sb&trk=ef359388-b2c8-4403-bbc0-768ca2834bfb&c_channel=e1

³ Source: <https://www.marketsandmarkets.com/Market-Reports/ai-code-tools-market-239940941.html>

Powered by IBM Granite® code models, watsonx Code Assistant delivers context-aware assistance in the IDE and through its integrated chat interface. It helps developers generate, complete, transform, explain, document, review, and test code efficiently.

The solution can be deployed on the cloud or on-premises and includes intellectual property indemnification, promoting trust and transparency in AI-assisted development.

watsonx Code Assistant features

IBM watsonx Code Assistant offers a range of features:

- ▶ Accelerated code generation

watsonx Code Assistant can generate code suggestions, unit tests, and complete functions or methods, enabling developers to work more efficiently and focus on higher-level tasks. Developers can write high-quality code that is aligned with established conventions by using simple, natural language prompts, regardless of experience level.

Figure 2-15 shows an initial prompt requesting a Python implementation of the quicksort algorithm.



Figure 2-15 watsonx request for Python code

Figure 2-16 displays the generated code and its accompanying description.

A screenshot of the watsonx Code Assistant interface showing the generated code and its description. The bot message says "Sure, here's a simple implementation of the quicksort algorithm in Python:". Below is the generated Python code:

```
# Assisted by watsonx Code Assistant
def quicksort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) // 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]
    return quicksort(left) + middle + quicksort(right)

# Test the function
print(quicksort([3,6,8,10,1,2,1]))
```

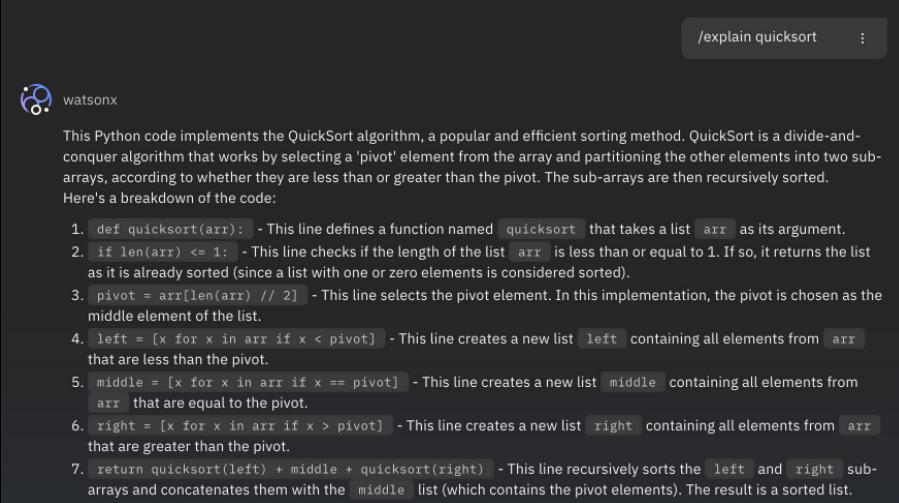
This program sorts an array by choosing a 'pivot' element from the array and partitioning the other elements into two sub-arrays, according to whether they are less than or greater than the pivot. The sub-arrays are then recursively sorted. Please note that this implementation uses list comprehensions for simplicity, which might not be the most efficient for large lists. For a more efficient implementation, you might want to use in-place partitioning. Also, the pivot is chosen as the middle element of the array in this implementation. Other strategies for choosing the pivot, such as picking a random element, can be used to improve performance on certain types of input. Lastly, this implementation is not stable, meaning that equal elements may not retain their original order. If stability is required, a different sorting algorithm should be used.

Figure 2-16 Response to request for Python code

► **Code explanation**

Using generative AI, watsonx Code Assistant analyzes and summarizes code to provide insights into its functions and purpose. This feature enhances transparency and supports better collaboration among development teams.

Figure 2-18 illustrates a code explanation that is generated by watsonx.



The screenshot shows the watsonx Code Assistant interface with the command `/explain quicksort` entered in the top bar. The main area displays the Python Quicksort algorithm and a detailed breakdown of each line:

```
def quicksort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) // 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]
    return quicksort(left) + middle + quicksort(right)
```

This Python code implements the QuickSort algorithm, a popular and efficient sorting method. QuickSort is a divide-and-conquer algorithm that works by selecting a 'pivot' element from the array and partitioning the other elements into two sub-arrays, according to whether they are less than or greater than the pivot. The sub-arrays are then recursively sorted. Here's a breakdown of the code:

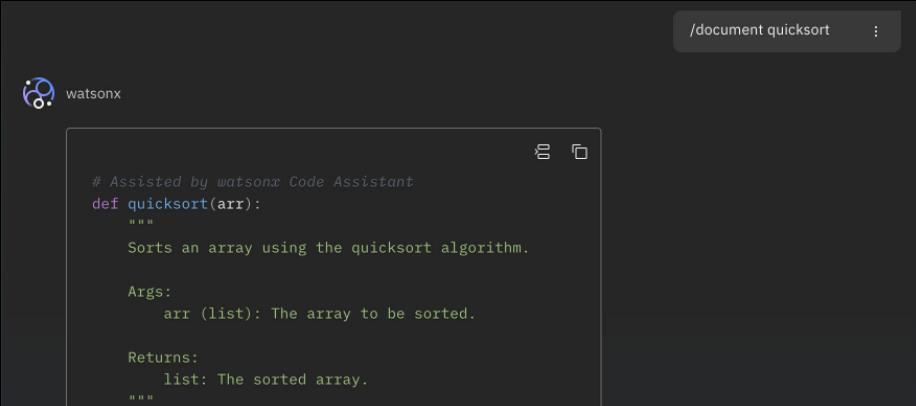
1. `def quicksort(arr):` - This line defines a function named `quicksort` that takes a list `arr` as its argument.
2. `if len(arr) <= 1:` - This line checks if the length of the list `arr` is less than or equal to 1. If so, it returns the list as it is already sorted (since a list with one or zero elements is considered sorted).
3. `pivot = arr[len(arr) // 2]` - This line selects the pivot element. In this implementation, the pivot is chosen as the middle element of the list.
4. `left = [x for x in arr if x < pivot]` - This line creates a new list `left` containing all elements from `arr` that are less than the pivot.
5. `middle = [x for x in arr if x == pivot]` - This line creates a new list `middle` containing all elements from `arr` that are equal to the pivot.
6. `right = [x for x in arr if x > pivot]` - This line creates a new list `right` containing all elements from `arr` that are greater than the pivot.
7. `return quicksort(left) + middle + quicksort(right)` - This line recursively sorts the `left` and `right` sub-arrays and concatenates them with the `middle` list (which contains the pivot elements). The result is a sorted list.

Figure 2-17 *Code explanation*

► **Code documentation**

watsonx Code Assistant can automatically generate comment lines that describe what the code does, making it simpler for developers to understand and maintain the codebase.

Figure 2-17 shows the comments that are added for documentation purposes.



The screenshot shows the watsonx Code Assistant interface with the command `/document quicksort` entered in the top bar. The main area displays the Python Quicksort algorithm with generated documentation comments:

```
# Assisted by watsonx Code Assistant
def quicksort(arr):
    """
    Sorts an array using the quicksort algorithm.

    Args:
        arr (list): The array to be sorted.

    Returns:
        list: The sorted array.
    """


```

Figure 2-18 *Generated code*

► **Enhancing code quality**

By generating unit tests and providing code explanations, watsonx Code Assistant helps ensure that applications are accurate, reliable, and maintainable.

2.6.2 IBM watsonx Code Assistant for Red Hat Ansible Lightspeed

IBM watsonx Code Assistant for Red Hat Ansible Lightspeed is designed to streamline and automate the Ansible development lifecycle, enhancing IT efficiency and scalability with trusted AI. It simplifies content creation through natural language Playbook generation, detailed task explanations, and a customizable model that delivers tailored recommendations, which empower developers of all skill levels.

Key features include the following ones:

- ▶ Playbook generation and code explanation
Reduce the time that is required to create complete Ansible Playbooks from natural language inputs. watsonx Code Assistant also provides detailed explanations for each task, improving understanding and confidence in the generated content.
- ▶ Model tuning
Customize the assistant by training it with your existing Ansible Playbook content. This feature enables more personalized and contextually relevant code recommendations that align with your enterprise standards.
- ▶ Ansible content generation
Simplify Playbook creation by using natural language descriptions in the Ansible Task field.
- ▶ Content source matching
Gain insight into the origin of code suggestions to make informed decisions about whether to accept or reject them.

2.6.3 Benefits of AI code assistants

The influence of AI on software development extends far beyond automating routine tasks. AI coding assistants, which are powered by ML, can learn from vast codebases to offer contextually relevant suggestions that align with a project's architecture, coding style, and best practices. You receive syntactically correct code through a more holistic development approach that streamlines workflows, enhances maintainability, and accelerates delivery.

Here are some of the benefits of AI code assistants:

- ▶ Boost developer productivity
One of the most immediate benefits of AI coding assistants is increased developer productivity. According to internal IBM testing, developers that use IBM watsonx Code Assistant reported:
 - Up to 90% time savings on code explanation
 - 59% reduction in documentation time
 - 38% reduction in code generation and testing time

Figure 2-19 shows an example of Java code generation that uses Watsonx Code Assistant.

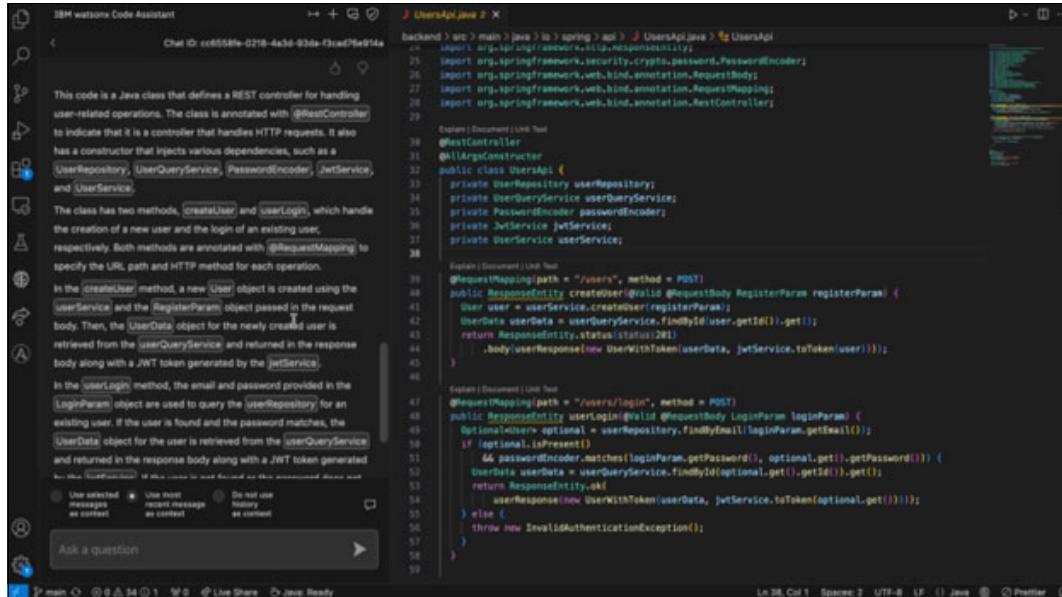


Figure 2-19 Java code that is generated by Watsonx Code Assistant

- ▶ Elevate code quality

IBM Watsonx Code Assistant plays a critical role in improving code quality. With real-time feedback and intelligent suggestions, it helps developers identify potential issues early in the development process. By promoting best practices and optimizing code structure, it reduces errors, enhances maintainability, and strengthens software reliability and security.

Poor code quality can lead to bugs, security vulnerabilities, and costly maintenance. AI-powered coding assistants help mitigate these risks by identifying issues early, enabling the development of robust, scalable, and maintainable software. Integrating AI-driven insights into the development workflow also helps reduce technical debt and accelerate delivery.

- ▶ Foster collaboration and knowledge sharing

AI coding assistants support collaboration, especially in distributed and remote development teams. Features such as real-time code suggestions, shared knowledge bases, AI-driven feedback loops, and output sharing enhance communication and alignment across teams. These capabilities help standardize code quality and practices, regardless of team members' experience levels or geographic locations. This consistency can help promote collaboration and accelerate the development process.

- ▶ Lower the barrier for new developers

AI-powered coding assistants also lower the barrier to entry for new developers. By automating routine tasks (such as code snippet generation, code completion, and documentation) and offering intelligent suggestions and reviews, tools like IBM Watson Code Assistant make it simpler for beginners to learn programming languages and contribute confidently.

With natural language processing (NLP) capabilities, these assistants provide real-time feedback, suggest improvements, and offer helpful resources, making the learning process more accessible and engaging.

Looking ahead, the potential of AI-driven development is immense. As AI technologies continue to evolve, coding assistants will become increasingly advanced and move beyond basic code suggestions to anticipate developer needs, understand project context, and provide proactive, intelligent assistance.

The growing trend of personalized learning will make AI coding assistants even more adaptive to individual developers. By tailoring suggestions and feedback based on a developer's style, preferences, and skill level, these tools will deliver a more customized, efficient, and enjoyable coding experience.



Client examples and use cases

A modernization journey on IBM Power servers typically follows a structured approach to enhance infrastructure, applications, and processes. The goal is to align with modern technologies, meet evolving IT needs, and support business objectives. As technology advances, organizations seek to integrate new capabilities, optimize performance, and future-ready their environments.

One common challenge at the beginning of this journey is that clients often lack clarity about their options and how to proceed. This chapter provides examples of how IBM clients, including IBM internal services, modernized their Power infrastructure and applications.

The following topics are described in this chapter:

- ▶ 3.1, “IBM Power modernization references” on page 88
- ▶ 3.2, “Banking solutions” on page 88
- ▶ 3.3, “IBM CIO: Hybrid by Design” on page 89
- ▶ 3.4, “CI/CD in Cirrus” on page 95
- ▶ 3.5, “Computer Systems Integration Ltd” on page 98

3.1 IBM Power modernization references

Companies across industries worldwide are gaining a competitive edge with technology running on IBM Power servers. By modernizing mission-critical applications on a cloud-native platform that uses Red Hat OpenShift and flexible capacity on demand, they can deliver industry-leading applications faster.

Improving IT infrastructure performance helps reduce energy consumption and supports a lower total cost of ownership (TCO) while aligning with green IT regulations. Implementing advanced isolation and integrity features enables ultra-secure, cloud-native, and containerized applications that reduce risk and accelerate hybrid cloud transformation.

Clients also benefit from a trusted pre-sales and post-sales support system that helps fast track successful co-creation, co-execution, and co-management.

For more information and to find examples of how choosing IBM Power leads to tangible outcomes, see this [IBM Power Modernization References document](#).

3.2 Banking solutions

The need to enhance customer experience while managing cost pressures, sustainability goals, regulatory compliance, and cybersecurity threats has led many banks to adopt hybrid cloud solutions to modernize their IT environments. Hybrid cloud enables banks to operate with greater agility, allowing them to scale quickly and respond to customer demand. It also supports the rapid adoption of technologies that improve efficiency, security, and sustainability.

IBM and Red Hat offer a robust hybrid cloud solution for banking modernization. By using Red Hat OpenShift on IBM Power servers, banks can build scalable, secure, and agile digital banking applications. This solution supports three key use cases:

- ▶ Using new cloud services:
 - Modernize to use real-time payments: Meet open banking regulations, enable faster payment methods, and simplify fund transfers across institutions.
 - Enhance anti-fraud activities: Access data quickly and apply adaptive rules to detect suspicious activity.
 - Drive business innovation: Use new tools to create offers, improve customer experiences, and grow the business.
- ▶ Incrementally modernize earlier applications: Modernize applications running on AIX and IBM i platforms.
- ▶ Extending core banking independent software vendor (ISV) applications: Support new digital engagement patterns.

Figure 3-1 on page 89 illustrates the IBM and Red Hat ecosystem that supports this modernization journey.

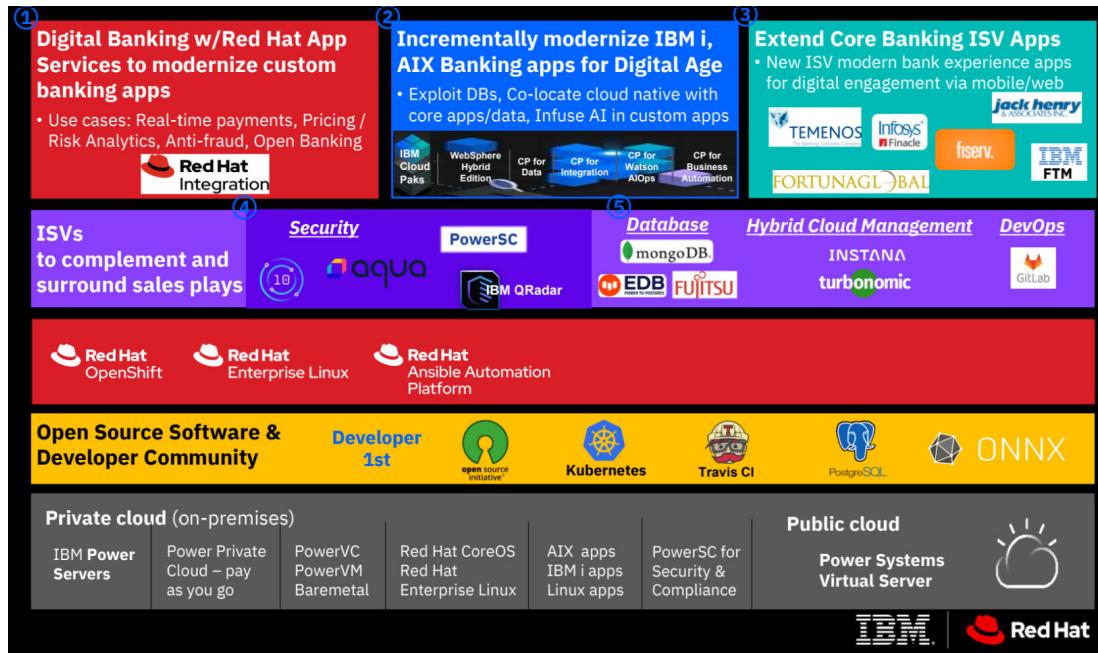


Figure 3-1 Banking use cases

For one approach to banking modernization, see [Banking and Finance Modernization with MongoDB and Red Hat OpenShift on IBM Power](#).

3.3 IBM CIO: Hybrid by Design

IBM built a world-class hybrid cloud platform that uses Red Hat OpenShift across IBM Cloud, IBM Power servers, and IBM Z mainframes. Over the past 2 years, IBM achieved more than \$3.5 billion in productivity gains, with \$2 billion attributed to artificial intelligence (AI) and automation.

IBM transformed its operating model by eliminating complexity, simplifying end-to-end workflows, automating manual tasks, and embedding AI into its hybrid cloud environment.

Central to this transformation is the principle of testing IBM technologies internally before offering them to clients. This approach is known as *Client Zero*.

3.3.1 IBM CIO Cirrus

Cirrus, IBM's internal hybrid cloud platform, is built by using Red Hat and IBM technologies. Designed to be modern, Cirrus delivers the speed, scale, security, and simplicity that are required for IBM's digital business experiences. It connects data, information, and people to help drive IBM's future.

The IBM CIO Hybrid Cloud serves as the engine of digital transformation for the IBM Chief Information Officer (CIO) organization. It delivers security-rich speed, scale, and simplicity, which are capabilities that you cannot achieve through public or private cloud alone. This platform enables IBM's digital application teams to focus on what matters most: advancing IBM's digital business.

Cirrus is guided by three core principles:

- ▶ Hybrid: Provides a consistent experience across all platforms.
- ▶ Integrated: Offers a unified continuous integration and continuous deployment (CI/CD) pipeline and operating environment.
- ▶ Open: Uses open-source and standards-based solutions.

At the heart of the IBM Hybrid by Design approach is the use of Red Hat OpenShift to create a common CI/CD pipeline that operates across x86, IBM Power (AIX, IBM i, and Linux), and IBM Z environments.

Figure 3-2 illustrates this approach.

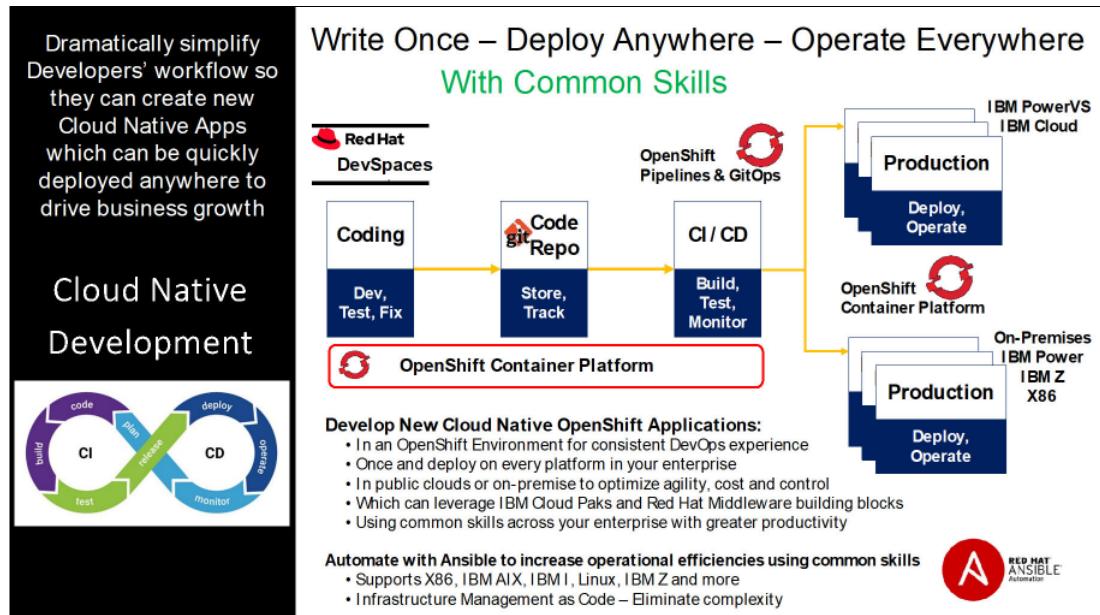


Figure 3-2 IBM CIO Cirrus goals

IBM is a global company with more than 250,000 employees and annual revenues exceeding USD 50 billion. Managing this diverse portfolio requires thousands of internal applications that support functions such as chip design, sales, marketing, management, accounting, customer support, and more. Some applications date back to the early days of computing, and others support cutting-edge technologies like quantum computing.

IBM's application teams faced several challenges:

- ▶ Managing over 5,000 applications across 74 data centers
- ▶ Supporting 250,000+ users in more than 170 countries
- ▶ Controlling hosting and operational costs
- ▶ Keeping systems current with the latest security fixes
- ▶ Meeting high availability and disaster recovery (HADR) objectives
- ▶ Complying with corporate security requirements
- ▶ Implementing minor changes that previously took months

IBM Client Zero success enabled the company to consolidate its hybrid cloud environment, by using AI and automation to transform enterprise productivity.

Hybrid by Design

The IBM CIO organization envisioned a “Hybrid by Design” cloud platform, which is a single platform on which digital business application components run with full observability, transparency, and optimized cost for performance.

“The choice was clear: We must leverage IBM’s products and technology at enterprise-scale. For us this meant using IBM’s hybrid cloud technology to build an intelligent application platform to run internal applications, integrations, digital workflows and data components,” – Matt Luteson, CIO, VP Technology Platforms Transformation at IBM.

Figure 3-3 illustrates the IBM CIO vision

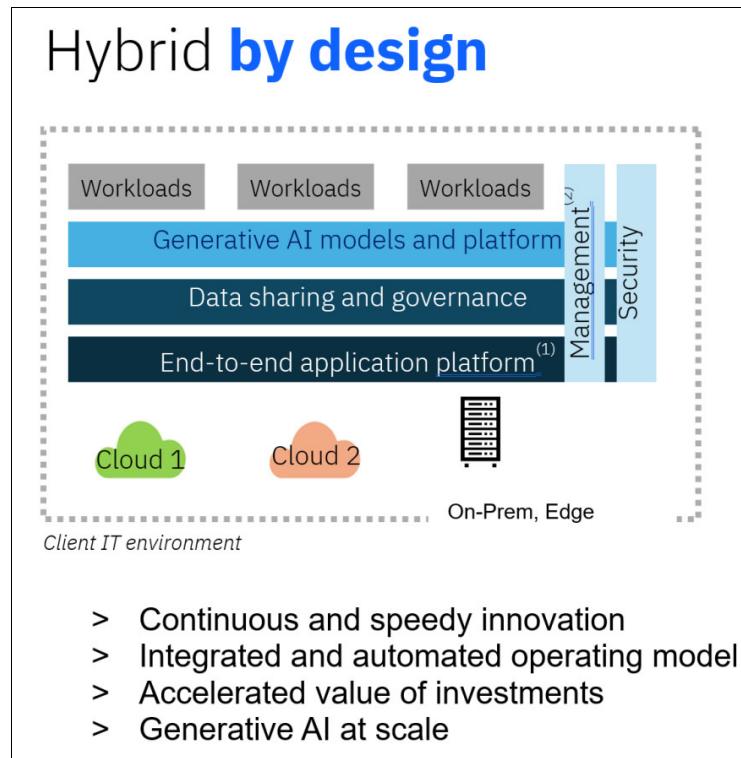


Figure 3-3 Hybrid by Design concepts

Platform engineering for CIO Hybrid Cloud at scale

A development team developed a unified hybrid cloud platform that they called the CIO Hybrid Cloud. This platform serves as a hosting environment for both containerized and virtual machine (VM)-based applications across a heterogeneous set of architectures. By using Red Hat OpenShift on IBM Cloud, the team can deploy identical environments both on-premises and in the cloud.

Figure 3-4 shows the CIO Hybrid Cloud architecture.

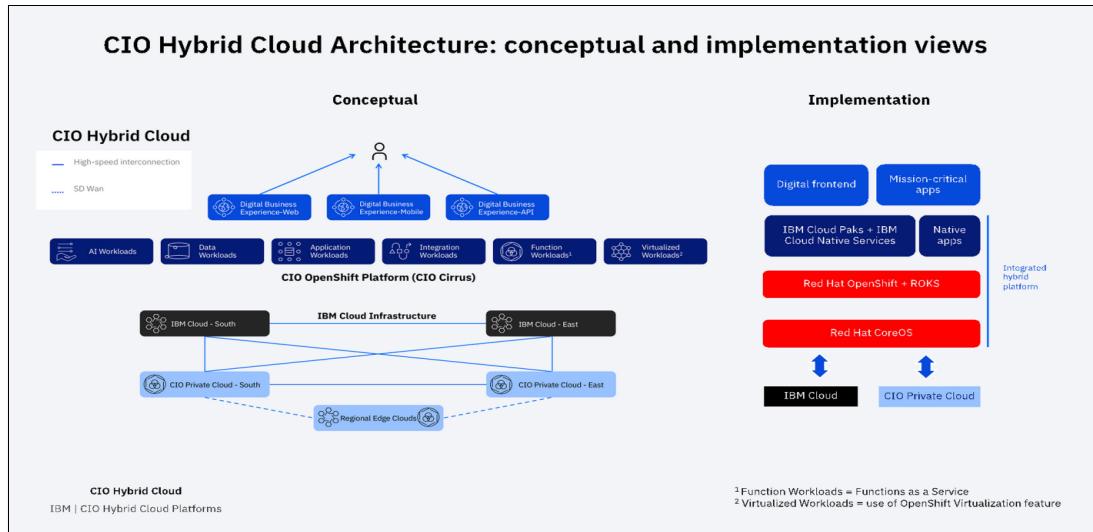


Figure 3-4 CIO Hybrid Cloud architecture

With increasing business demands around security, resiliency, and cost optimization, it became essential to deliver a hybrid cloud platform that enables IBM developers to focus on building enterprise applications for their stakeholders. The CIO Hybrid Cloud platform helps meet these requirements by serving as the hosting environment.

A set of IBM Cloud services that are integrated into the hybrid cloud platform provides the correct balance between application innovation and operational guardrails. These services support key areas such as databases, observability, and storage.

Figure 3-5 shows the hybrid cloud implementation.

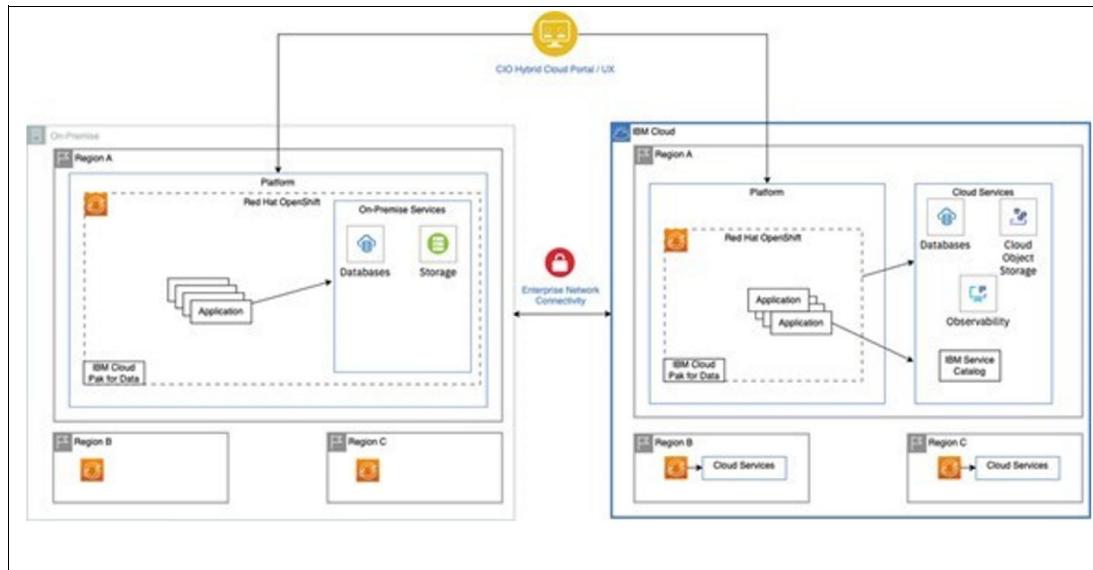


Figure 3-5 Hybrid cloud implementation

The expansion of the Red Hat OpenShift based Cirrus platform to Power servers was part of the Intelligent Workload Placement (IWP) initiative. This strategy enables Red Hat OpenShift applications to run seamlessly across multiple hardware architectures, including x86, Power, and IBM Z. It also accelerates application modernization by supporting the migration from monolithic applications to microservices and other modern development and deployment techniques.

Red Hat OpenShift on Power began gaining traction around 2017 - 2018 as IBM and Red Hat collaborated to enable the platform. In 2019, Red Hat announced Extended Update Support (EUS) for Red Hat OpenShift on Power, allowing clients to remain on a specific version longer with enterprise-grade support.

Later in 2019, IBM launched Power Virtual Server (PowerVS), an infrastructure as a service (IaaS) solution that simplified automation and accelerated client adoption. In 2020, Red Hat OpenShift 4.6 was released with new capabilities for Power, including support for compute nodes with up to 512 threads and enhanced storage options. Red Hat OpenShift 4.6 also became available simultaneously on IBM Z, IBM Power, and x86 platforms, which reinforced Red Hat OpenShift as a truly platform-neutral solution.

The IBM CIO Power Virtualization Service delivers end-to-end virtualization on IBM Power servers, maximizing availability, standardization, and automation. The team manages all on-premises virtualized Power server infrastructure across CIO data centers, providing strategic building blocks for a best-in-class solution. Key responsibilities include:

- ▶ Ongoing management and configuration of the global Power server infrastructure, including Virtual I/O Server (VIOS), Hardware Management Consoles (HMCs), PowerVC, storage, network virtual local area networks (VLANs), and zone architecture
- ▶ Adopting industry best practices and using IBM Consulting expertise to educate Application Hosting Service Squads
- ▶ Consolidating and standardizing support processes, tools, and monitoring
- ▶ Proactively developing automated solutions to improve global support efficiency
- ▶ Implementing transparent maintenance practices to eliminate planned outages and reduce the impact of unplanned ones
- ▶ Applying IBM Security® processes to maintain infrastructure health and compliance
- ▶ Tracking and addressing all security vulnerabilities that are related to Power servers
- ▶ Maintaining open collaboration between Physical and Automation squads

The IBM Power Squad successfully delivered a Red Hat OpenShift cluster on Power infrastructure that was integrated with Cirrus in one quarter. This integration enabled Cirrus to support workload migration from public to private cloud and to modernize applications running on AIX and Red Hat Enterprise Linux (RHEL).

Benefits that are gained

Red Hat OpenShift on Power helps optimize infrastructure costs by reducing the number of servers that are required without compromising performance. Figure 3-6 illustrates the TCO impact of using IBM Power compared to x86 servers with Red Hat OpenShift.

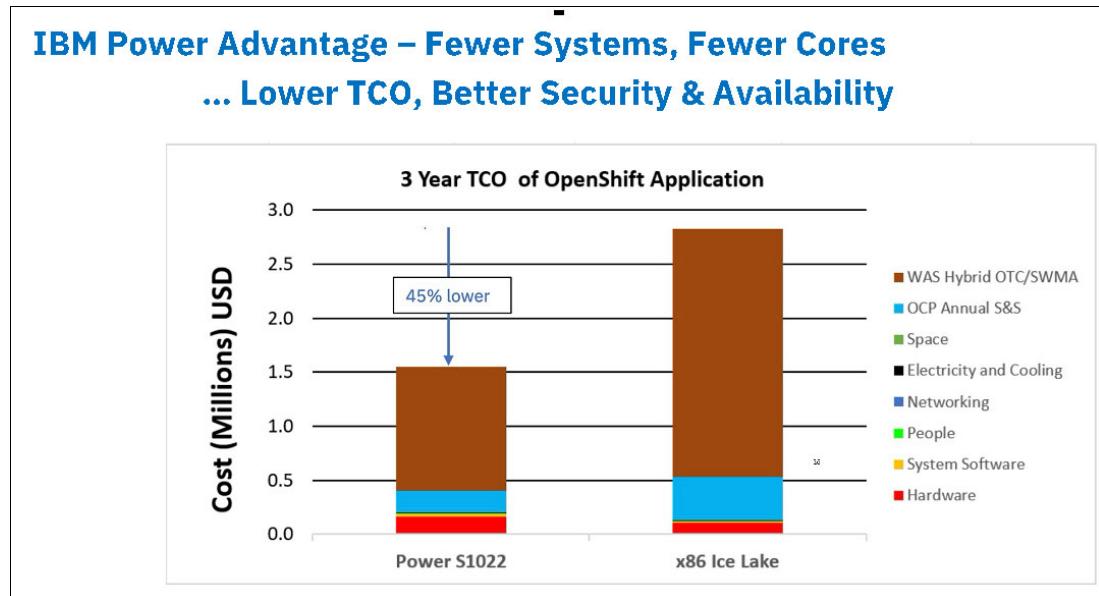


Figure 3-6 TCO benefits of Power versus x86 for Red Hat OpenShift

The CIO Hybrid Cloud platform delivered measurable results across IBM's enterprise:

- ▶ Over 2,000 application components were onboarded, including custom-built applications, integrations, and data components
- ▶ 55% reduction in DevOps hours, allowing application teams to focus on delivering business value
- ▶ 90% cost savings for hosting containerized workloads compared to earlier environments

Application teams are now meeting key corporate objectives:

- ▶ Accelerated change delivery: Application updates that once took months are now delivered daily. Security fixes are applied rapidly.
- ▶ Built-in security and availability: Secure, HA applications are now the default, rather than an afterthought.
- ▶ Integrated disaster recovery (DR): DR support is embedded in platform implementation patterns, eliminating the need for separate, costly solutions.

The increasing scale and capabilities of the previous private cloud solution required capital expenditure (CapEx) and high operational impact that were challenging to predict. The agility of the IBM hybrid cloud allowed the platform to scale elastically based on demand. The IBM Cloud Catalog of services is used by the platform and by the applications. The cloud provides security-rich resources with low operational impact and consumption-based pricing. Data and applications are on-premises or in the cloud based on business requirements. Cloud data residency can be regional to meet regulatory requirements.

Continuous platform improvement is based on data-driven decisions with measurable results. Real-time data that is generated by the applications is stored in a data warehouse and are available for platform and application analysis. Eventually, some applications require updates, and platform metrics inform the required changes. Other applications reach end of life and the impact of retiring an application can be assessed. Application owners have visibility into metrics through a common interface to assess performance and troubleshoot problems. The [IBM Cloud Pak for Data](#) is a central component of the data warehouse.

The platform can also analyze underlying performance and measure the impact of changes. The data is also used to report derived information to the application teams, including compliance posture, criticality, and software currency. The [IBM Cloud Security and Compliance Center](#), [IBM Cloud Log Analysis](#), and [IBM Cloud Activity Tracker](#) are used as part of the observability strategy along with IBM Business Partners.

The platform can apply AI-based analysis on the data warehouse. Vertical scaling of application hosting resources by using the [IBM Turbonomic](#) solution is available to the applications. IWP is also possible, allowing applications to migrate for optimal cost and performance on x86, IBM Power, or IBM Z platforms, such as the following items:

- ▶ 90% cost savings by using hybrid cloud containerized workloads
- ▶ 55% operations reduction with fewer platform operations resources
- ▶ Significant reductions in applications and software licenses

Application success by default

The CIO Hybrid Cloud hosts applications that are distributed across both private and public cloud environments. Each application environment begins with a “batteries-included” template that incorporates IBM’s business rules. A consistent application runtime environment is replicated across all data centers, enabling applications to run locally and maintain availability during regional disruptions. Applications run on compute and storage resources that are tailored to their requirements and consume cloud catalog services with consumption-based pricing.

Migrating to a hybrid cloud containerized environment reduced hosting expenses. Traditional VMs were replaced with containerized workloads. Operational responsibilities that are specific to individual applications shifted to a common platform, allowing teams to focus on delivering business value while reducing operational impact.

The IBM CIO Hybrid Cloud, when combined with application modernization and migration, delivers end-to-end observability and consistent speed, scale, and security.

The platform continues to evolve. Compliance requirements change, and application performance demands are increasing with the adoption of AI. Further optimization is possible by analyzing real-time application and cost data to determine the most effective placement of workloads across private and public cloud environments.

3.4 CI/CD in Cirrus

CI/CD is a set of practices that enables development teams to frequently merge code changes into a central repository and confidently deploy them to users.

As part of its hybrid cloud journey, the IBM CIO organization identified safe, scalable application development and deployment as a critical component of transformation. To support this goal, the organization is building a common, enterprise-wide CI/CD solution that is based on Red Hat OpenShift Pipelines. This standardized pipeline approach allows teams to build on shared best practices and scale them across the organization.

But why focus on a common CI/CD solution? The following sections outline the rationale and benefits of adopting a unified approach to CI/CD across the enterprise.

3.4.1 What CI/CD is

CI/CD is a two-step process that streamlines development and delivery through automation. CI involves developers frequently committing code changes to a shared repository. Each commit triggers an automated build and a series of tests to verify application behavior and help ensure clean integration of updates.

This automation enables teams to deploy code frequently, leading to CD, which automates the release of applications to production. Because it eliminates manual gates, CD relies heavily on robust test automation. Once an application passes all tests, a developer's change can be deployed to production within minutes.

3.4.2 How CI/CD is implemented

CI/CD automates the integration, release, and deployment of software, removing friction from the development process. It enables smaller, more manageable code changes, continuous testing, real-time feedback, faster releases, improved customer satisfaction, and reduced costs:

- ▶ Smaller code changes: CI/CD encourages frequent integration of small code updates. These changes are simple to manage and test, especially in large teams working on shared codebases.
- ▶ Continuous testing: Each code change is automatically tested on check-in, allowing developers to detect and resolve issues early.
- ▶ Real-time feedback: CI/CD provides continuous feedback to both developers and stakeholders, increasing transparency and accountability across teams.
- ▶ Faster releases and improved customer satisfaction: Automated pipelines help ensure consistent builds, tests, and deployments, which enable faster delivery of new features and bug fixes, allowing teams to respond quickly to user feedback.
- ▶ Reduced costs: By finding bugs early and automating deployments, CI/CD reduces the cost of fixing issues and minimizes the risk of human error.

3.4.3 A security example

In late 2021, a zero-day vulnerability was discovered in the widely used logging library Log4j 2. This vulnerability allowed attackers to exploit systems by submitting a simple string of text that triggered the execution of malicious code. It was rated at the highest possible Common Vulnerability Scoring System (CVSS) severity level and was estimated to affect more than 90% of enterprise cloud environments.

Beyond the immediate security risks, the incident revealed a deeper issue: Many organizations lacked visibility into the components that made up their systems or even which systems were running. It underscored the importance of understanding software composition and the associated security requirements.

What does this situation have to do with CI/CD? The Log4j 2 incident highlighted the need for enterprises to embed guardrails into their development and deployment processes. A standardized CI/CD pipeline can provide this visibility and control, helping organizations strengthen their security posture through automation, traceability, and consistent policy enforcement.

Security guardrails

Recent high-profile vulnerabilities prompted enterprises and chief security officers to rethink their security strategies. As a result, many organizations are adopting new practices:

- ▶ Secrets and credentials detection
- ▶ Open-source library inventories
- ▶ Open-source usage approvals
- ▶ Open-source vulnerability detection
- ▶ Code quality and automated test coverage
- ▶ Static Application Security Testing (SAST)
- ▶ Dynamic Application Security Testing (DAST)
- ▶ Container image vulnerability detection
- ▶ Artifact signing

These activities often introduce new compliance requirements for development teams. CI/CD pipelines offer a natural starting point for integrating these security practices. By embedding security into the development process, CI/CD helps teams discover issues early and maintain consistent security standards.

A common CI/CD pipeline makes this integration even simpler. It allows new security tools to be added without disrupting existing workflows, enabling developers to focus on delivering business value while protecting the enterprise.

This approach is the one that IBM is taking with its common CI/CD journey. By integrating security into the CI/CD process, IBM is introducing guardrails that achieve the following goals:

- ▶ Enforce security policies, including automated scanning.
- ▶ Reduce developer response time to vulnerabilities like Log4j.
- ▶ Promote application quality and consistency.
- ▶ Offload compliance and maintenance burdens.

On the CD side, the platform also supports the following tasks:

- ▶ Auditing of production deployment approvals
- ▶ Integration of DAST processes
- ▶ Tracking of deployment frequency and duration
- ▶ Visibility into deployment region and location

Economy of scale and friction reduction

Most development teams recognize the value of CI/CD pipelines, but adoption can be hindered by barriers such as cost, skill gaps, and time constraints. As compliance requirements grow, these barriers become even more pronounced, introducing friction that can reduce developer productivity.

From an organizational perspective, allowing each team to implement its own CI/CD approach creates inefficiencies. Teams must individually build and maintain pipelines, manage infrastructure, and integrate new tools, which result in increased costs and delays in feature delivery.

A common CI/CD pipeline addresses these challenges. A centralized team manages pipeline creation, infrastructure support, and tool integration. For development teams, this approach leads to the following outcomes:

- ▶ Greater focus on delivering business value
- ▶ Increased productivity and developer satisfaction
- ▶ Accelerated application modernization

This shared approach reduces the burden of CI/CD adoption and minimizes development friction. By using a unified infrastructure, organizations benefit from economies of scale.

The IBM CIO organization adopted this model to collect and analyze pipeline data. A centralized data lake aggregates information from a common set of tools, providing visibility from source code to deployment. A developer experience portal offers a single pane of glass for understanding and acting on this data.

This approach enables IBM to achieve the following goals:

- ▶ Better understand open source usage.
- ▶ Improve visibility into application quality.
- ▶ Gain insights into run times and programming languages.
- ▶ Reduce response times to security issues, such as Log4j.

CI/CD is a set of practices that enables teams to safely build and deploy applications. A common CI/CD solution scales these benefits across the enterprise while reducing adoption effort. It also allows security and compliance activities to be integrated into pipelines without overwhelming development teams. which results in a safer, more informed organization and more satisfied developers.

3.5 Computer Systems Integration Ltd

Computer Systems Integration Ltd is a managed service and integration provider that is based in the UK and operates in the US under the name Tectrade. A Platinum IBM Business Partner for over 40 years, the company specializes in IBM Power servers, high-performance computing (HPC), storage, and high-end vendor integration solutions.

Figure 3-7 shows the Computer Systems Integration Ltd Innovation Lab, which is a result of a long-term collaboration between Computer Systems Integration Ltd and IBM. Initially created to showcase IBM's vision for enterprise hybrid cloud and modernization solutions, the lab evolved to support pre-general availability (pre-GA) offerings, Early Adoption Programs, and first-of-a-kind hybrid cloud, modern, and AI solutions.

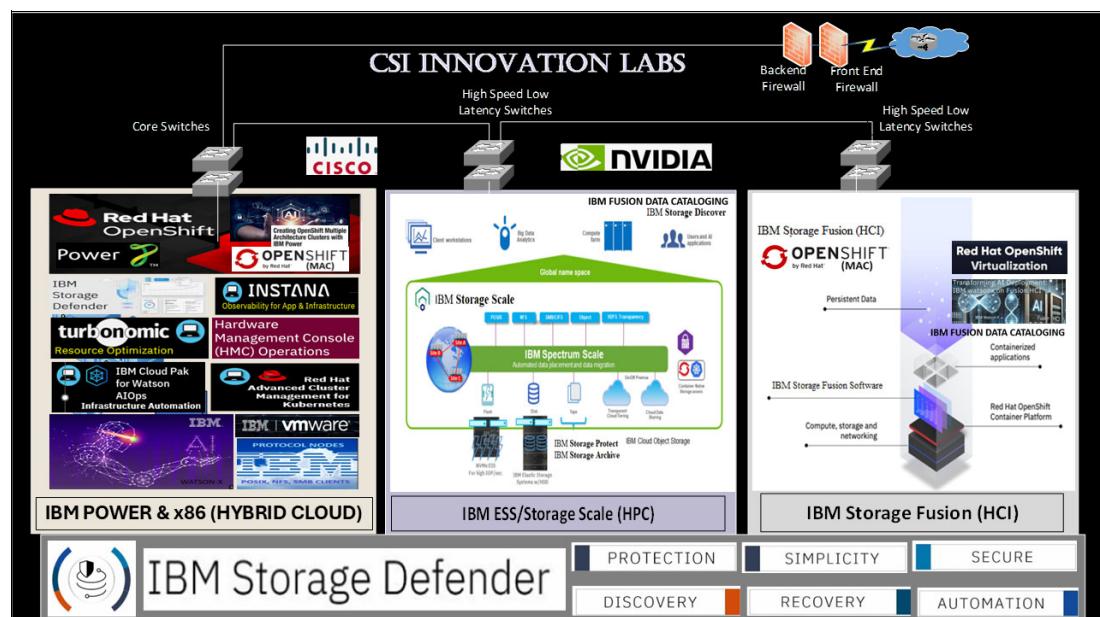


Figure 3-7 Diagram of the Computer Systems Integration Ltd Innovation Lab environment

The Innovation Lab demonstrates how enterprise IBM Power servers integrate seamlessly with other hybrid cloud vendor solutions. This unique environment serves as a demonstration platform for hybrid cloud innovation.

The solution showcases IBM's extensive technology portfolio, with a strong focus on using AI to address practical use cases. It integrates seamlessly with other vendor technologies to deliver both "art of the possible" demonstrations and production-ready solutions. The solution is organized into three primary categories:

- ▶ Hybrid Cloud Core

IBM Power10 series and x86 platforms form the foundation of the hybrid cloud environment, enabling high-performance, scalable, and secure workloads.

- ▶ IBM Storage Scale

The Storage Scale System family supports HPC workloads with scalable, high-throughput storage solutions.

- ▶ IBM Fusion Hyper-Converged Infrastructure (HCI)

This component supports AI workloads, Red Hat OpenShift virtualization, and containerized environments. It includes built-in backup and restore capabilities, along with IBM Fusion Data Cataloging for enhanced data management.

3.5.1 IBM Power and x86 CIO Hybrid Cloud components

The CIO Hybrid Cloud solution enables customers to explore the integration of IBM Power and x86 architectures within a unified environment. It uses industry-leading technologies to deliver scalable, secure, and intelligent operations. Key components include the following ones:

- ▶ Red Hat OpenShift
- ▶ IBM AI on IBM Power
- ▶ Instana on Power
- ▶ Turbonomic
- ▶ Red Hat Advanced Cluster Management
- ▶ IBM Cloud Pak for AIOps Infrastructure Automation
- ▶ Advanced automation tools

Figure 3-8 shows the CIO Hybrid Cloud components.

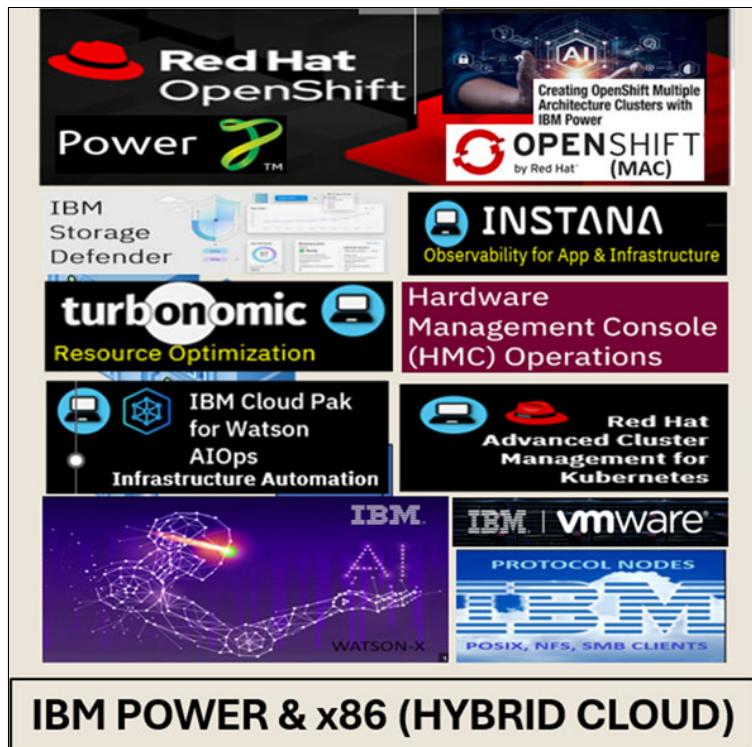


Figure 3-8 CIO Hybrid Cloud components

The CIO Hybrid Cloud provides a workspace that enables clients to gain insights into optimizing infrastructure for modern workloads, improving observability and enhancing resource management.

Key technologies that are covered

The CIO Hybrid Cloud provides visibility into key technologies that support enterprise modernization and hybrid cloud operations:

- ▶ Red Hat OpenShift and Kubernetes
Deploy multi-architecture Red Hat OpenShift clusters with IBM Power and x86 worker nodes.
- ▶ Red Hat Advanced Cluster Management
Manage and govern multiple hybrid cloud Red Hat OpenShift clusters through a single interface.
- ▶ IBM Cloud Pak for Watson AIOps Infrastructure Automation
Enable AI-driven infrastructure automation and anomaly detection.
- ▶ Turbonomic Resource Optimization
Automatically optimize workload placement and cost efficiency.
- ▶ Instana Observability
Monitor applications and infrastructure in real time.
- ▶ HMC
Apply best practices for managing IBM Power servers.

- ▶ IBM and VMware Integration
Streamline hybrid cloud operations with seamless compatibility.
- ▶ IBM watsonx.ai®
Unlock AI-driven insights and automation across cloud environments.
- ▶ IBM Storage Defender
Strengthen security and data protection in hybrid cloud deployments.

3.5.2 IBM Storage Scale and IBM Storage Fusion

As part of the Innovation Lab, the CIO Hybrid Cloud environment is integrated with IBM Elastic Storage System (ESS) and IBM Storage Scale and IBM Storage Fusion, highlighting their roles in supporting modern AI, analytics, and enterprise workloads. The solution emphasizes IBM Spectrum Scale for scalable, high-performance data management, and IBM Storage Fusion for containerized environments running on Red Hat OpenShift, offering integrated data services for modern applications

Figure 3-9 shows the storage components that are connected to the CIO Hybrid Cloud.

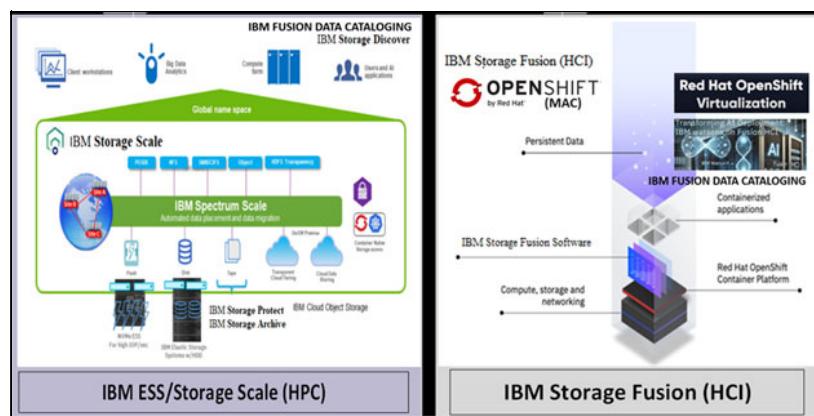


Figure 3-9 Storage components in the CIO Hybrid Cloud

When used with Red Hat OpenShift Multi-Architecture Compute capabilities, the CIO Hybrid Cloud environment enables seamless movement of applications (Pods) between cluster nodes running on x86 and IBM Power servers. Instana provides real-time observability and monitoring, and integrates with Turbonomic to automate workload management and improve cost efficiency. This capability is unique to Red Hat OpenShift Multi-Architecture Compute with both IBM Power and x86 worker nodes.

To meet the growing demand for scalable, high-performance storage, IBM offers two distinct architectures:

- ▶ **IBM ESS and IBM Storage Scale (HPC)**
A high-performance, scalable storage solution that is built on IBM Storage Scale (formerly known as IBM Spectrum Scale), and designed for automated data placement and migration in HPC environments.
- ▶ **IBM Storage Fusion (HCI)**
A hyper-converged infrastructure solution that uses Red Hat OpenShift and IBM Storage Fusion software to support persistent storage, containerized applications, and AI workloads.

IBM Storage Scale

IBM Storage Scale provides a global namespace, automated data management, and cloud integration to deliver a highly scalable, enterprise-level storage solution. IBM Storage Scale Systems (formerly known as ESS) offer a purpose-built hardware implementation of IBM Storage Scale (formerly known as IBM Spectrum Scale and General Parallel File System (GPFS). These systems help ensure high IOPS and flexible storage configurations, supporting both Nonvolatile Memory Express (NVMe) and HDD options with policy-driven data placement.

For more information, see [IBM Storage Scale](#).

IBM Storage Fusion

IBM Storage Fusion is a software-defined, AI-driven infrastructure solution that is designed for Red Hat OpenShift environments. It includes built-in IBM Fusion Data Cataloging (formerly known as IBM Spectrum Discover) and supports backup and restore for both containers and VMs. IBM Storage Fusion is available as a software-only offering or as part of an appliance-based solution.

Red Hat OpenShift Virtualization

Red Hat OpenShift Virtualization enhances AI deployment by supporting containerized storage solutions and providing a platform for migrating VMs from high-cost environments.

Key use cases include the following ones:

- ▶ AI and machine learning (ML) workloads that use IBM Watsonx on IBM Storage Fusion HCI
- ▶ High-performance computing (HPC) with IBM Storage Scale Systems and IBM Storage Scale
- ▶ Cloud-native storage for containerized applications
- ▶ Data protection and archiving with IBM Storage Protect and IBM Storage Archive



Services and consulting options

Modernization projects are often complex and carry a high level of risk. Engaging consulting services for infrastructure modernization provides businesses with a strategic advantage by offering access to specialized expertise and experience. These projects typically involve intricate technologies and complex implementations. Consultants help mitigate risks and reduce costly errors through their in-depth knowledge and up-to-date understanding of industry best practices.

Consultants help develop comprehensive modernization strategies that are aligned with specific business goals. They assess current infrastructure and create detailed implementation roadmaps. In addition, they play a critical role in risk mitigation and cost optimization by identifying potential disruptions and recommending efficient solutions to avoid unnecessary expenses.

By using partnerships with technology vendors, businesses gain access to a broader range of resources and solutions, helping ensure the selection of appropriate technologies. Ultimately, outsourcing infrastructure modernization enables businesses to focus on their core competencies. This approach frees internal resources and maintains productivity while consultants manage the technical aspects of the project, helping ensure a smooth and efficient transformation.

This chapter describes some of the services that IBM and IBM Business Partners offer to support clients in their modernization journey.

The following topics are described in this chapter:

- ▶ 4.1, “IBM Business Partners” on page 104
- ▶ 4.2, “IBM Technology Expert Labs” on page 106
- ▶ 4.3, “Client Engineering” on page 107
- ▶ 4.4, “IBM Consulting” on page 108

4.1 IBM Business Partners

IBM has a comprehensive ecosystem of IBM Business Partners who collaborate to deliver solutions and services across various industries. These partnerships are essential for extending IBM's reach, providing specialized expertise, and developing innovative offerings.

This ecosystem includes a wide range of organizations, such as solution providers, system integrators, managed service providers, independent software vendors (ISVs), and consultancies, all working together to use IBM's advanced technologies.

4.1.1 IBM Partner Plus Directory

Whether you are an IBM Business Partner or a company seeking support, you can use the IBM Partner Plus® Directory to find IBM Business Partners who specialize in services and modernization methods on IBM Power. At the time of writing, a search for the phrase "modernization on Power" returns 1,877 IBM Business Partners worldwide.

Figure 4-1 shows the IBM Partner Plus Directory home page.

The screenshot displays the IBM Partner Plus Directory homepage. At the top, there is a search bar with the query "modernization on power". Below the search bar, there are two tabs: "Partners" (selected) and "Solutions". On the left, there is a sidebar with filters for "Partner plus tier" (Silver, Gold, Platinum), "Partner type", "Country", "Diverse revenue business", and "Reseller authorization". The main content area shows a grid of partner cards. The first row contains four cards: "IT-Power Services GmbH" (Wien, AT), "IT-Power Services Deutschland GmbH" (Lüneburg, DE), "Power Maroc" (CASABLANCA, MA), and "IT-Power Services Emilia-Romagna Srl" (Italy). The second row contains three cards: "Rocket Software, Inc." (Westham, US), "Certus Solutions Limited" (Auckland, NZ), and "Certus Solutions Pty Ltd" (Melbourne, AU).

Figure 4-1 IBM Partner Plus Directory

IBM Business Partners deliver IBM-certified solutions that are supported by IBM Services® worldwide. This approach helps ensure that technologies and solutions are communicated and implemented in a way that is clear and accessible to the client.

For example, IBM collaborates with its IBM Platinum Business Partner, Saudi Business Machines (SBM), in the Middle East to deliver modernization solutions on IBM Power servers. In this partnership, SBM provides the infrastructure, and IBM works with ISVs and the client to deliver software solutions by using IBM Cloud Pak for Integration.

In a typical IBM Power modernization scenario, upgrading the infrastructure is often the first step. This upgrade provides the compute resources and virtualization layer that are needed to support modernization. Local IBM Business Partners offer the cohesion and expertise that are required to move these projects forward successfully.

Note: To find for a certified IBM Business Partner, search the [IBM Partner Plus Directory](#).

Figure 4-2 shows an example of a modernization project that includes IBM Power with IBM Cloud Pak for Integration (CP4I).

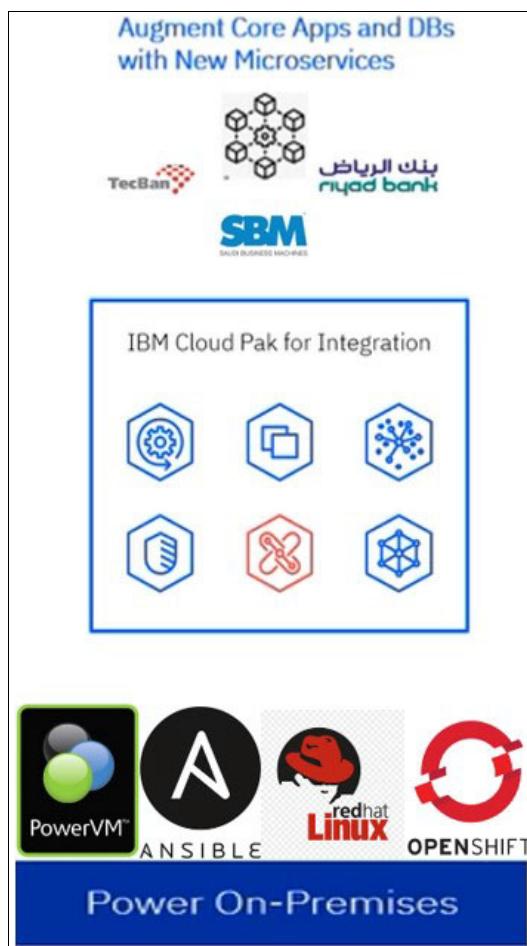


Figure 4-2 Example of modernization project components

4.2 IBM Technology Expert Labs

IBM Technology Expert Labs is a professional services organization that is staffed by a global team of experienced product experts. This team brings deep technical knowledge across both software and infrastructure, including IBM data and artificial intelligence (AI), automation, sustainability, security, software-defined networking, IBM Power, IBM Storage, IBM Z and LinuxONE, IBM Z software, IBM GDPS, and IBM Cloud.

The mission of IBM Technology Expert Labs is to accelerate the adoption (deployment, consumption, and expansion) of IBM's hybrid cloud and AI strategy. They achieve this goal through expert guidance, best practices, and seamless digital experiences that drive continuous business outcomes.

Infrastructure consultants from IBM Technology Expert Labs deliver services either remotely or onsite, offering deep technical expertise, valuable tools, and methodologies. With a global presence, the team can deploy experienced consultants worldwide. These experts have worked with large enterprises across industries such as finance, insurance, healthcare, and retail to support modernization initiatives on IBM Power.

IBM Technology Expert Labs offers several compelling reasons to engage their services:

- ▶ A best practice methodology: Consultants apply established methodologies, practices, and patterns to help achieve better business outcomes.
- ▶ Specialized industry experts: The team develops complex solutions and minimizes implementation risks through deep industry expertise.
- ▶ Direct ties to development: Close connections with IBM product development provide early access to product insights, features, and solutions.
- ▶ Accelerated adoption: Consultants help clients realize value faster from IBM infrastructure through offerings that span the entire project lifecycle. These offerings include assessment, planning, implementation, migration, upgrade, and post-implementation advisory services.

IBM consultants support a wide range of modernization use cases on IBM Power:

- ▶ Planning and deployment of Red Hat OpenShift on IBM Power for IBM Power10 Private Cloud Rack with Db2 Warehouse.
- ▶ Application modernization for banking through Red Hat OpenShift on IBM Power.
- ▶ Modernizing IBM i by using the IBM i Modernization Engine for Lifecycle Integration (Merlin) on Red Hat OpenShift to enable continuous integration and continuous deployment (CI/CD).
- ▶ Migrating workloads to IBM Power Virtual Server on IBM Cloud for AIX, IBM i, and SAP HANA.
- ▶ Automating provisioning and day-2 operations, such as fix management and logical partition (LPAR) management, by using HashiCorp Terraform and Red Hat Ansible.
- ▶ Building IBM PowerVC environments to create an IBM Private Cloud infrastructure for IBM Power servers, reducing provisioning time and optimizing virtual machine (VM) management.
- ▶ Implementing Shared Utility Capacity to enable a cloud-like consumption model.

For more information about pre-scooped and pre-priced Standard Offerings for faster solution implementation in your organization, see [IBM Technology Expert Labs for Systems Standard Offerings](#).

IBM's subscription-based advisory offering, Expertise Connect, augments your team with deep technical expertise through flexible engagement options. Whether you need project-based assistance or on-demand support for ongoing operations, Expertise Connect helps you achieve your goals efficiently.

You can [Partner with IBM Technology Expert Labs](#) and use IBM Services in migrating your on-premises IBM Power servers to IBM Power Virtual Server in IBM Cloud. For more information, see [IBM Technology Expert Labs infrastructure services](#). For IBM employees and IBM Business Partners, see [Systems Expert Labs](#).

4.3 Client Engineering

Client Engineering is an IBM investment that is designed to foster joint innovation and rapidly validate solutions for business challenges by using IBM Hybrid Cloud and AI technologies, as illustrated in Figure 4-3.



Figure 4-3 Client Engineering mission

- ▶ What does IBM offer to clients?

IBM provides a no-cost, multidisciplinary team with deep technical expertise to co-create and rapidly validate solutions that are tailored to specific business needs. These teams use IBM technologies to deliver impactful outcomes.

- ▶ What value do clients receive?

Clients gain confidence in the technical solutions addressing their business challenges and benefit from accelerated time to value.

- ▶ How does IBM deliver this value?

IBM uses its user-centric Pilot Engineering Method to innovate, iterate, and validate solutions. Through this approach, IBM deploys more than 1,700 experts in multidisciplinary squads to co-create with clients by using accelerators to streamline solution development.

What a pilot is

A *pilot* is a rapid co-creation engagement that demonstrates the value of IBM technologies in achieving a client's business outcomes. Using a user-centric approach, the pilot identifies high-impact use cases, defines the scope, contributes to the business case, and builds a solution that meets the client's requirements.

By working closely together, IBM and the client build confidence in the solution, foster a collaborative partnership for long-term success, and establish a foundation for future development and scaling.

Pilot Engineering Method

Figure 4-4 illustrates the Pilot Engineering Method that IBM uses during client engagements. The process begins with solution workshops to define requirements and align the solution with business objectives. IBM then collaborates with the client to build the solution and prepares the environment for migration, helping ensure a smooth migration.

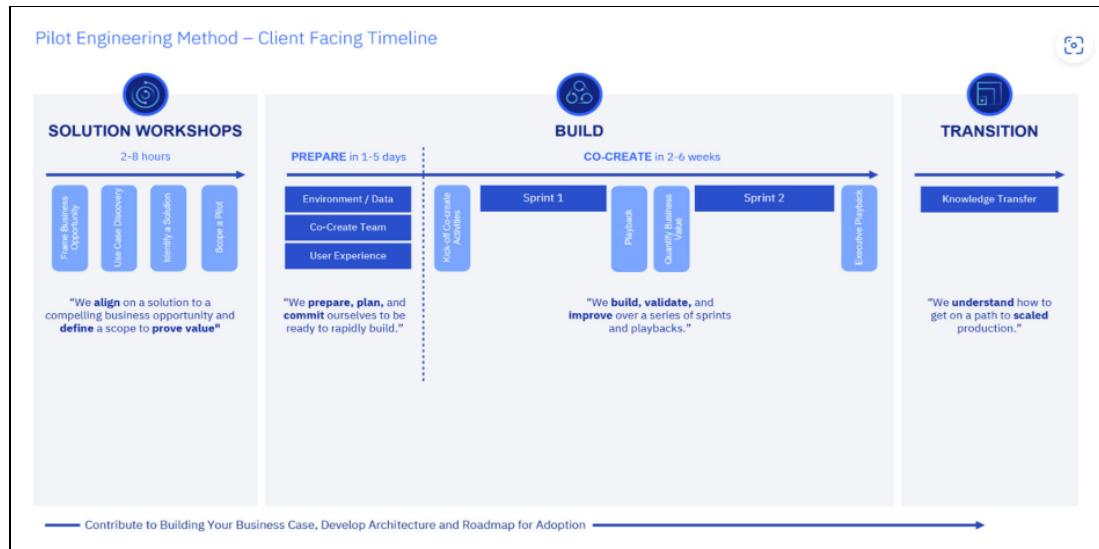


Figure 4-4 Pilot Engineering Method

Skills and capabilities

IBM Client Engineering EMEA includes teams of subject matter experts (SMEs) with extensive experience in IBM Power servers. These teams conduct pilots within Power infrastructure environments, optimizing the client experience.

IBM Client Engineering proposes pilots that are focused on technologies that are critical to IBM Power clients:

- ▶ Power and AI
- ▶ Cloud migration with IBM Power Virtual Server
- ▶ Automation by using Red Hat Ansible, IBM PowerVC, and HashiCorp Terraform
- ▶ Code modernization with IBM Watson Code Assistant™
- ▶ Application modernization through containerization with Red Hat OpenShift

IBM Client Engineering EMEA also collaborates closely with TechZone to deliver custom environments for pilots involving IBM software solutions. These environments are tailored to meet specific infrastructure needs, client standards, and non-functional requirements.

How to engage

To engage with Client Engineering, contact your IBM or IBM Business Partner Sales representatives.

4.4 IBM Consulting

IBM Consulting (formerly known as IBM Global Business Services®) is the professional services and consulting division of IBM. It serves a broad range of clients, including enterprises, government agencies, non-profits, and NGOs, by providing expertise to advise, design, build, and operate business innovation initiatives.

With a significant global footprint, IBM Consulting operates offices and deploys consultants in numerous countries. Its headquarters is in Armonk, New York, with major offices in locations such as Yorktown Heights and San Francisco in the US, Hursley in the UK, and Bangalore in India. This global presence enables IBM to support clients across diverse geographies and industries.

IBM Consulting provides a range of assets and productivity tools that are designed to enhance efficiency and drive business value. These assets fall into two broad categories:

- ▶ Business value assets

These assets include software products and platforms that are typically retained by the client after an engagement. Business value assets are listed in the Asset Catalog and are market-priced. Licensing options include subscription and perpetual models. IBM continues to onboard new business value assets and enhance traditional ones through the Advantage program.

- ▶ Productivity-based assets and assistants

These tools are designed to improve the productivity of IBM consultants. Although primarily used by IBM teams, clients may also engage with these tools during collaborative work.

Use cases

Standard use cases in the software development lifecycle often focus on accelerating productivity and modernization. IBM supports these use cases through advanced tools and AI-driven assistants:

- ▶ Accelerating business rule extraction from code
- ▶ Optimizing ongoing development and modernization activities
- ▶ Assisting Java-to-microservice transformation by using IBM Watsonx Code Assistant
- ▶ Achieving a 30% - 40% productivity increase during development through generative AI assets and assistants
- ▶ Generating pseudocode and performing framework-to-framework conversions
- ▶ Creating test case scenarios, performance test plans, and preparing test data

IBM Consulting offers a range of services to support software quality and modernization efforts:

- ▶ Test case and automation creation: Generate detailed test cases from project requirements and create automated testing scripts that are compatible with various tools and programming languages.
- ▶ Mainframe and system support: Develop test cases that are tailored for systems to help ensure continued reliability and performance.
- ▶ Data structuring and code conversion: Transform unstructured test documentation into organized formats and generate diverse test datasets.
- ▶ Performance testing and reporting: Create comprehensive test plans and reports to detect issues early and optimize system performance.
- ▶ AI-powered quality checks: Use AI to compare test results with expected outcomes, improving testing accuracy and reducing the risk of production defects.

IBM Consulting serves as a strategic partner for organizations seeking to address complex business challenges and drive innovation and growth. By using IBM's extensive technological resources and deep industry expertise, IBM Consulting helps clients navigate transformation and achieve meaningful outcomes.



Modernizing the management of IBM Power servers

Modernizing management capabilities for IBM Power servers is essential to meet the demands of a digital-first world, where businesses must operate with agility, scalability, and security. User interfaces (UIs) and dashboards play a critical role in effective system management. IBM provides interfaces such as the IBM Power Hardware Management Console (HMC) and IBM Navigator for i. However, as the number of systems increases, it becomes important to integrate automation and artificial intelligence (AI)-driven tools to deploy applications at scale across the enterprise. The goal is to upgrade hardware and evolve management and monitoring tools to support modern workloads efficiently.

Tools such as IBM Power Virtualization Center (PowerVC) (IBM's virtualization management product, which is based on OpenStack), Ansible, and Terraform provide automation capabilities for managing IBM Power servers. Automation simplifies system management as environments grow. It enables rapid integration of new servers while enforcing standards and helping ensure that systems remain secure and efficient.

These management advancements are especially important as environments expand to hybrid models, including cloud offerings such as IBM Power Virtual Server, which allows workloads to run on IBM Power servers in the IBM Cloud. Also, IBM Power Private Cloud with Shared Utility Capacity helps optimize costs by using shared resources and a pay-per-use, by-the-minute pricing model for compute capacity in Power Enterprise Pools 2.0 (PEP2).

The following topics are described in this chapter:

- ▶ 5.1, “HMC simplification” on page 112
- ▶ 5.3, “Power Virtualization Center overview” on page 118
- ▶ 5.4, “Ansible OpenStack Modules” on page 135
- ▶ 5.6, “Power Enterprise Pools and Cloud Management Console” on page 171
- ▶ 5.7, “PowerVM networking concepts” on page 173

5.1 HMC simplification

The HMC is an appliance that is used to plan, deploy, and manage IBM Power servers. It enables the creation and modification of logical partitions (LPARs), including the dynamic addition and removal of resources for running partitions.

Over the past few years, IBM continuously improved the HMC interface by adding features that simplify operations and enhance automation. Many functions that previously required manual configuration in the Virtual I/O Server (VIOS) or the HMC command-line interface (CLI) can now be performed through the GUI.

5.1.1 System and partition templates

System and partition templates contain details for the system or partition resources, including processor count, memory allocation, physical adapters, virtual networks (VNets), and storage settings. Users can quickly deploy servers or create LPARs by using quick-start templates available in the template library or custom, user-defined templates that are stored in the HMC, as shown in Figure 5-1.

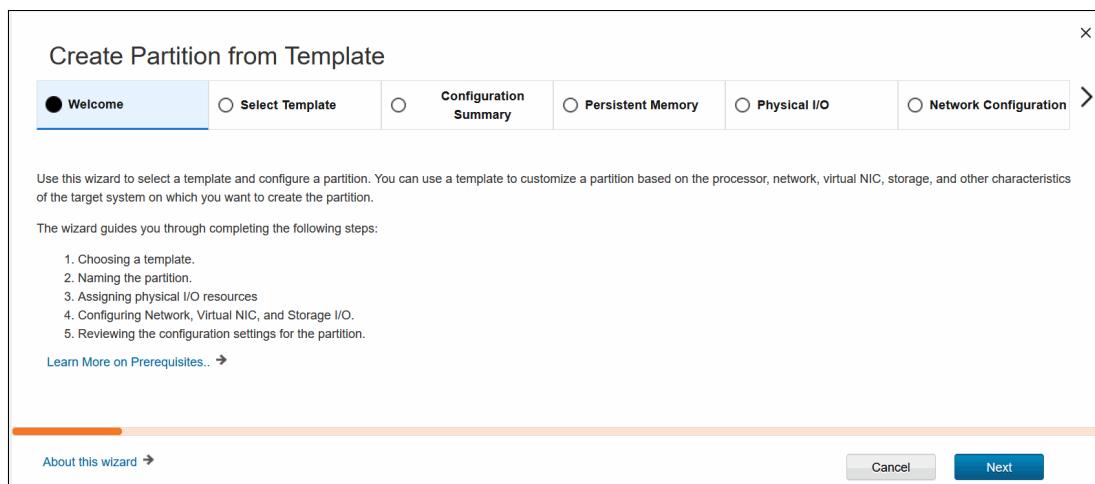


Figure 5-1 Partition template wizard

5.1.2 Shared Ethernet Adapter simplification

The Shared Ethernet Adapter (SEA) bridges internal network traffic to a physical network adapter, eliminating the need for each client LPAR to have a dedicated physical adapter for external network access.

The HMC provides a function to create an SEA through the GUI, eliminating the need to use the VIOS CLI. The interface includes a wizard that supports various SEA configurations and allows users to modify features such as failover, virtual local area network (VLAN) tagging, load sharing, and Etherchannel. With the GUI, users can adjust VLANs or modify SEA parameters without using the CLI.

This process is illustrated in Figure 5-2 on page 113.

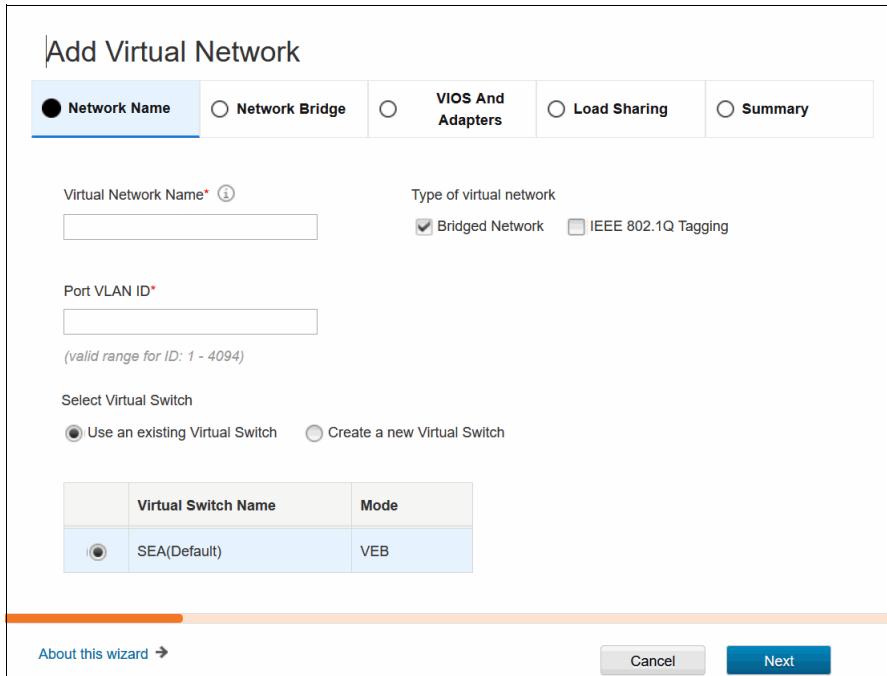


Figure 5-2 Shared Ethernet Adapter creation wizard

5.1.3 Hardware Management Console update and upgrade

Keeping the HMC firmware at a supported release level is essential to help ensure the security and stability of the entire IBM Power server infrastructure. The latest HMC firmware releases include enhancements that simplify both the update and upgrade processes.

HMC update

The HMC update process automatically retrieves all prerequisites along with the selected update and installs them in the required order. All necessary fixes are downloaded directly from the IBM website.

HMC upgrade

The HMC upgrade process is enhanced to automate and streamline the upgrade workflow. The process includes the following steps:

1. Save the HMC upgrade data.
2. Download the upgrade files.
3. Restart and upgrade the HMC.

5.2 Virtual I/O Server management improvements

VIOS plays a critical role in PowerVM virtualization by allowing physical I/O adapters to be virtualized and shared across multiple virtual machines (VMs) on a server. VIOS operates as an LPAR running a specialized version of the operating system (OS) that is designed for efficient resource sharing. To help ensure redundant access to shared adapters, a best practice is deploying two VIOS LPARs per server.

IBM maintains and periodically updates VIOS, introducing new features and addressing known issues. It is the client's responsibility to keep VIOS updated to maintain security and minimize downtime.

Recent VIOS releases simplified maintenance by integrating upgrade functions into the HMC GUI. This automation reduces upgrade time and minimizes the risk of errors. This section describes the new functions.

5.2.1 Updating VIOS

The HMC interface introduces a function to update the VIOS directly through the GUI. A VIOS update installs new service packs within the same technology level.

Previously, VIOS updates required that you use the VIOS CLI. As part of PowerVM simplification, VIOS update images can now be imported from a remote Network File System (NFS) or sFTP server, through a USB device, or downloaded directly by the HMC from the IBM website. The entire process is fully automated and managed through the HMC GUI, as shown in Figure 5-3.

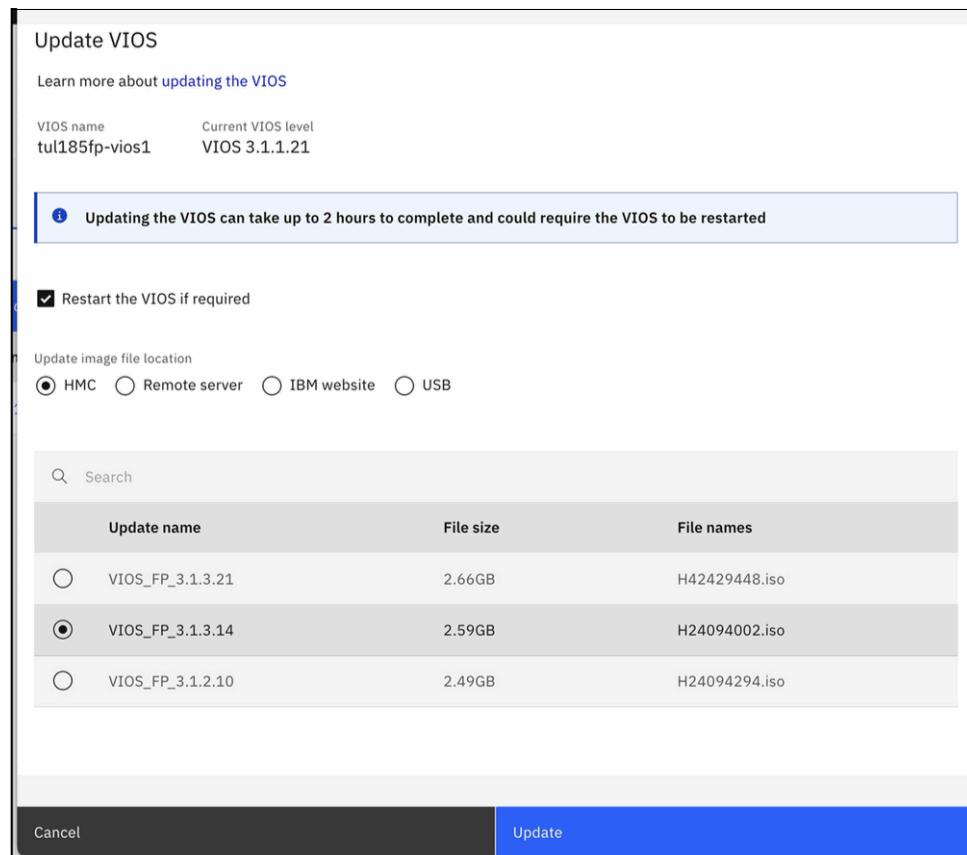


Figure 5-3 VIOS update wizard

To learn more about the function, see this [IBM blog](#).

5.2.2 Upgrading VIOS

A VIOS upgrade is required when moving between different technology levels (for example, from Version 3.0.x to 4.0.x). Previously, this process required using the **viosupgrade** tool from the VIOS CLI. As part of PowerVM simplification, the entire upgrade process can now be managed through the HMC GUI.

The HMC provides a wizard that guides users through each step of the upgrade and manages all necessary components, streamlining the process and reducing the potential for errors.

Figure 5-4 shows the VIOS upgrade wizard.

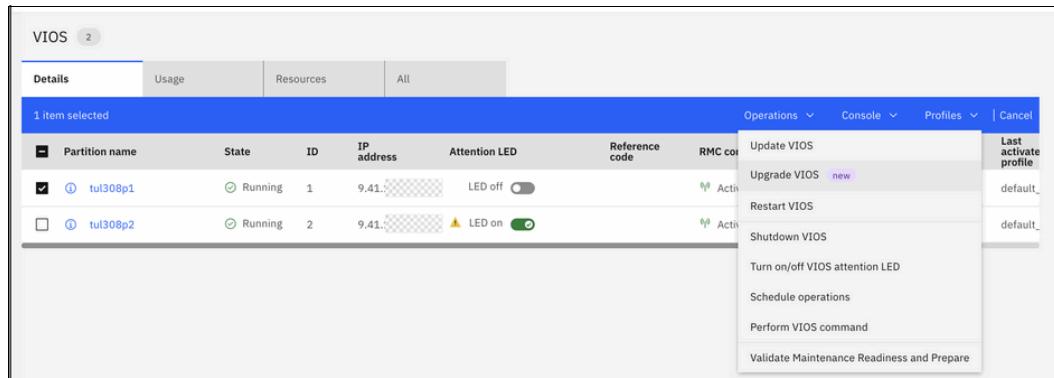


Figure 5-4 VIOS upgrade wizard

To learn more about this function, see this [IBM blog](#).

5.2.3 Automated Virtual I/O Server backups

The HMC interface enables fully automated backups of the VIOS. As part of PowerVM simplification, the HMC provides a function to schedule and perform VIOS backups. Backup images can be stored in the HMC repository for immediate use in a restore operation or offloaded to a remote location.

From the GUI, users can schedule either a full VIOS backup (similar to the **backupios** command) or a backup of only the I/O configuration (similar to the **viosbr** command). This function is illustrated in Figure 5-5.

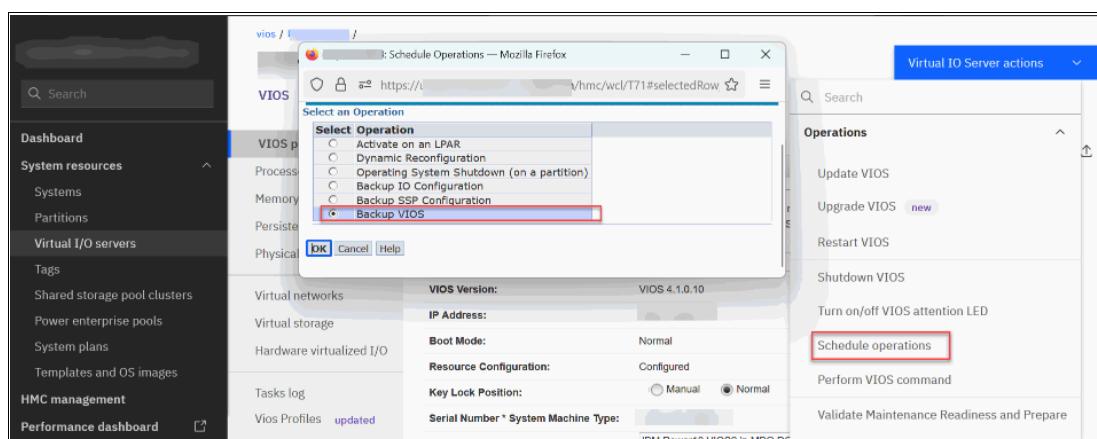


Figure 5-5 HMC schedule operation: Creating a VIOS backup

5.2.4 Restoring VIOS

The I/O configuration of the VIOS can be restored with the HMC GUI but restoring a VIOS backup is not supported at the time of writing. It can be achieved from the HMC CLI with `installios` command.

You can restore the I/O configuration of the VIOS by using the HMC GUI. However, restoring a full VIOS backup is not supported through the GUI. This operation must be performed by using the HMC CLI by running the `installios` command.

For more information about this function, see [Capturing, Importing, Exporting, and Restoring VIOS backups on the HMC](#).

5.2.5 Validating the maintenance readiness of the Virtual I/O Server

During VIOS maintenance, it might be necessary to power off a VIOS LPAR. This action can impact client partitions that rely on the VIOS for storage or network connectivity. Therefore, before powering off a VIOS, users should validate the redundancy of the dual-VIOS configuration to help ensure that client partitions are properly configured with multipath and redundant access to network and storage devices.

The enhanced HMC interface provides a wizard that guides users through this validation process, helping ensure system resilience during maintenance operations.

For more information about this function, see [Prepare VIOS for Maintenance and other PowerVM management enhancements](#).

5.2.6 Installing Virtual I/O Server images from the HMC

As part of PowerVM simplification, VIOS images can now be stored in the HMC repository. These stored images can be used to install VIOS directly from the HMC interface. Images can be imported from a remote location, DVD media, or a USB device.

Figure 5-6 shows the VIOS image wizard.

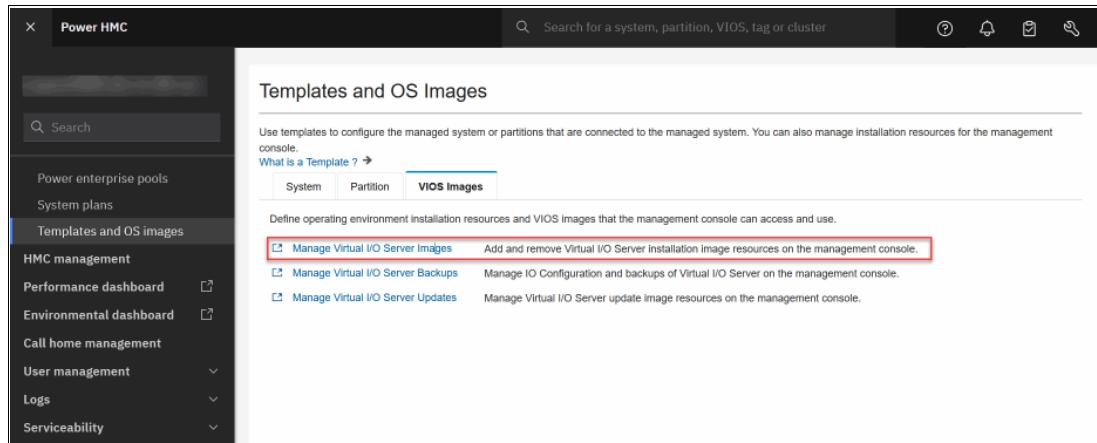


Figure 5-6 Virtual I/O Server image wizard

5.2.7 Microcode code update in I/O adapters

Although VIOS is a critical component of the PowerVM environment, it is equally important to plan for regular microcode updates to the I/O adapters that are associated with the VIOS configuration.

As part of PowerVM simplification, microcode updates can now be performed directly from the HMC GUI. The HMC wizard automatically identifies the latest available microcode levels from the IBM website and provides a list of adapters that require updates.

Figure 5-7 shows the simplified microcode upgrade function.

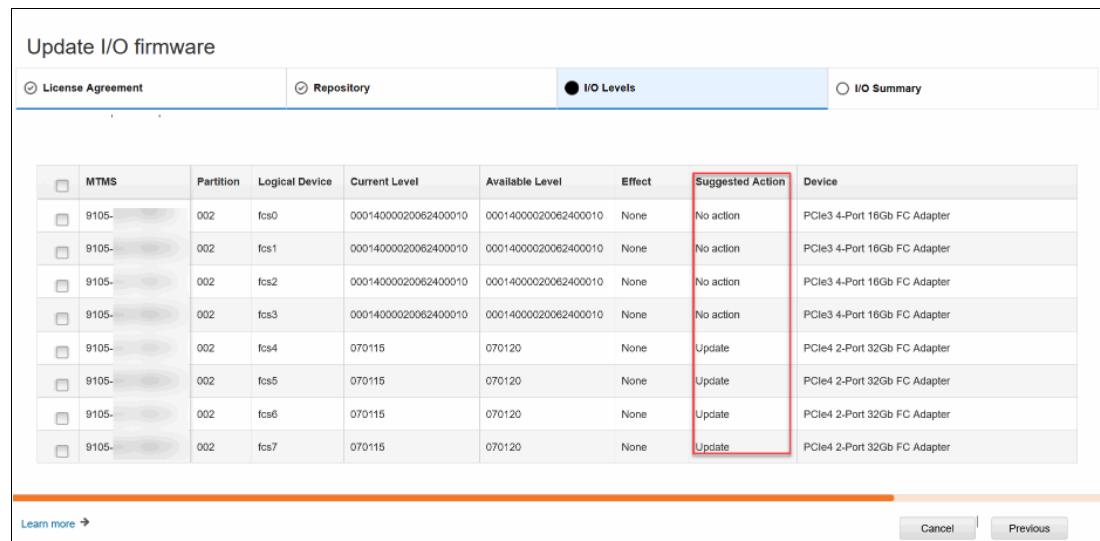


Figure 5-7 Microcode update wizard

5.2.8 NFS-mounted ISOs in the Virtual Media Library

The VIOS 4.1.1.00 Fix Pack introduces support for mounting ISO images from an NFS server, making them available as installable media within the Virtual Media Library.

This enhancement allows ISO images to be loaded directly from a centralized NFS server, eliminating the need to copy ISO files across multiple VIOS instances. This approach saves storage space, reduces deployment time, and helps ensure consistency across environments. It supports both NFS v3 and v4 and allows multiple ISO images to be linked into the repository.

The `mkvopt` command is enhanced with the new `-nfslink` option, which creates a symbolic link to the specified NFS-hosted ISO file within the repository. This function is demonstrated in Example 5-1.

Example 5-1 Mounting NFS ISO file

```
$ mkvopt -name <image_name_in_VML_repository> -file /mnt/<mounted_ISO_file.iso>
-nfslink -ro
```

For more information about this new capability, see this [IBM PowerVM Community blog](#).

5.3 Power Virtualization Center overview

PowerVC is a robust cloud and virtualization management solution that is designed for IBM Power servers. It streamlines the management of VMs, simplifying tasks such as creation, deployment, resizing, and migration. PowerVC enables organizations to build and manage private cloud environments on an existing Power infrastructure, offering on-demand resource allocation and self-service provisioning.

Built on OpenStack, PowerVC provides a flexible and scalable architecture that integrates seamlessly with other OpenStack-based tools and services. By simplifying cloud deployments and optimizing workload and policy management, PowerVC helps reduce total cost of ownership (TCO) and maximize resource usage while delivering an intuitive user experience.

PowerVC captures and manages essential infrastructure details, including VM definitions, storage, networking, and server configurations. IT teams can maintain a library of VM images, enabling rapid deployment by using preconfigured images rather than manually rebuilding environments. Centralized image management accelerates the migration and deployment of virtual images across available systems.

The solution also allows administrators to create resource groups to support workloads, enhancing operational efficiency by quickly adapting to workload demands. This flexibility reduces the administrative processing impact and enables IT departments to respond more effectively to business needs and market trends.

Figure 5-8 shows how PowerVC can improve your client experience.

The screenshot displays the PowerVC management interface. At the top, a banner reads "PowerVC: Complete lifecycle management for VMs". Below it, a sidebar on the left contains a summary of PowerVC's capabilities: "PowerVC provides comprehensive virtualization and private cloud management for Power servers and integration with multi-cloud managers". The main area is divided into three sections: "Virtual Machines", "Hosts", and "Dashboard". The "Virtual Machines" section shows a table with columns: Name, Host, IP, State, Health, Owner, and Expiration Date. The "Hosts" section lists hosts named "nec10", "nec17", "nec18", "nec19", "nec20", "nec21", "nec22", and "nec23", each with details like Virtual Machines, Processors, Memory (GB), State, and Health. The "Dashboard" section provides a high-level overview with four circular metrics: 160 hosts (75% healthy), 500 VMs (90% healthy), 90% memory usage, and 1,000 storage units (60% used).

Figure 5-8 PowerVC management capabilities

Here are some key benefits of PowerVC:

- ▶ Seamless installation and configuration of the entire hardware stack, including host, storage, network components, and software.
- ▶ Rapid setup, with installation that is typically completed within hours, regardless of the administrator's skill level.

- ▶ Small footprint and streamlined client experience across the full lifecycle, from deployment to ongoing operations and support.
- ▶ A reliable, cost-effective, and extensible platform for delivering virtualization and cloud capabilities on IBM Power servers.
- ▶ Industry-standard APIs that enable integration with higher-level cloud services.
- ▶ Efficient workload management through policy-based optimization, dynamically adjusting resources or migrating workloads to underutilized systems.

For more information about PowerVC, see [Introduction to PowerVC for Private Cloud](#).

PowerVC functions

IBM PowerVC can perform the following functions with IBM PowerVM:

- ▶ Dynamically deploy VM OpenStack images with storage and network resources.
- ▶ Dynamically resize VM CPU and memory by using custom compute templates (OpenStack flavors).
- ▶ Dynamically assign disk volumes from predefined storage connectivity groups.
- ▶ Dynamically assign new network interfaces from predefined networks.
- ▶ Import existing VMs that are visible to the HMC for management by IBM PowerVC.
- ▶ Monitor and manage resource usage by using project namespaces and quotas.
- ▶ Create a snapshot or clone an existing VM image for deployment or backup.
- ▶ Migrate VMs between managed systems by using Live Partition Mobility (LPM).
- ▶ Remotely restart a VM on a different Power managed system if the host fails.
- ▶ Use advanced storage technologies, such as VDisk mirroring or IBM Global Mirror.
- ▶ Improve resource usage to reduce capital expenditure (CapEx) and power consumption.

PowerVC uses both the IBM PowerVM Dynamic Partitioning abilities of dynamic logical partitioning (DLPAR) and the Open Source Cloud management abilities of OpenStack to control and manage large pools of compute, storage, and networking resources, all managed through APIs or a dashboard.

PowerVC offerings

PowerVC is built on OpenStack and provides simplified virtualization management and cloud deployment capabilities for IBM AIX, IBM i, and Linux VMs running on IBM Power servers. The offering is designed to enable private cloud capabilities on Power servers and improve administrator productivity. PowerVC can also integrate with broader cloud environments by using higher-level cloud orchestrators.

PowerVC is available in two editions:

- ▶ 5765-VCC IBM Cloud PowerVC Manager
- ▶ 5765-VC2 PowerVC for Private Cloud

The primary difference between the editions is that PowerVC for Private Cloud enables self-provisioning for project users. The most recent version of PowerVC is 2.3.x.

Figure 5-9 shows the recent release history for PowerVC at the time of writing. For the most current information, see [PowerVC Lifecycle Information](#).

PowerVC Release	PowerVC intermediate Release	Latest service pack	Release Date	End of Service Pack Support (EoSPS)	End of Support (EOS)
2.3.x Standard 3 years support	2.3.0		12 Dec 2024	30 Sep 2025	
	2.3.1		25 July 2025	30 Sep 2026	30 Apr 2028
	2.3.2		TBD	TBD	

Figure 5-9 PowerVC 2.3 service history

Note: For more information about what is new in IBM PowerVC for Private Cloud, see [What's new in PowerVC for Private Cloud](#).

5.3.1 PowerVC architecture

PowerVC can be deployed in multiple configurations, depending on your requirements.

Single-node configuration

The single-node deployment of IBM PowerVC, also known as the All-in-One (AIO) model, consolidates all PowerVC components, including OpenStack-based and stateful services, onto a single physical system or VM. This model simplifies installation and management, making it suitable for smaller environments or proof-of-concept deployments.

Although the AIO model offers reduced complexity, it inherently limits scalability and high availability (HA). A single point of failure can disrupt the entire PowerVC environment. Therefore, although the AIO model is convenient for initial setups, it becomes less suitable as the data center grows and requires greater resilience and capacity.

High availability configuration

Early versions of IBM PowerVC used an AIO deployment model, which consolidated all required packages and services, both stateless (OpenStack-based) and stateful, onto a single system. Although this approach simplified deployment and management, it limited the ability of PowerVC to scale and achieve HA, which is essential for larger data centers with multiple IBM Power servers.

To address these limitations, PowerVC evolved to support a multi-node architecture that improves both scalability and resilience. This architecture eliminates single points of failure by introducing a HA management node, helping ensure that the failure of one node does not disrupt virtualization and cloud management operations. Also, distributing workloads across multiple nodes enables PowerVC to support larger deployments. PowerVC 2.0.2 introduced a three-node architecture, establishing the foundation for a scalable and enterprise-ready solution.

Figure 5-10 on page 121 shows the HA configuration.

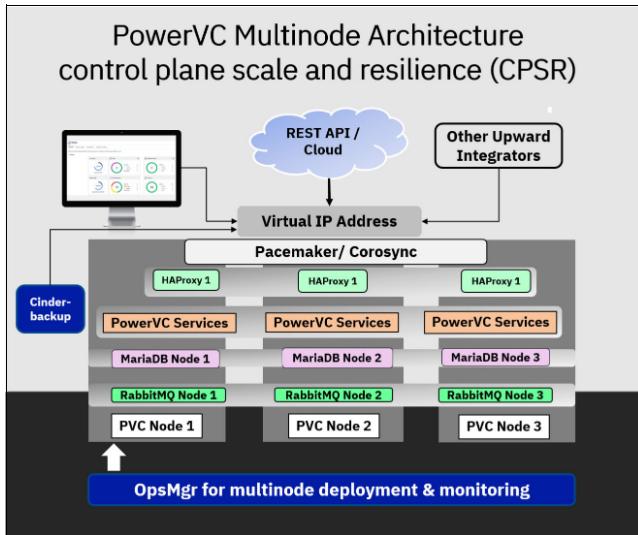


Figure 5-10 PowerVC multinode implementation

Compute nodes for more scalability

Compute plane nodes (CPNs) in PowerVC enhance scalability and reliability by distributing the compute services load, particularly when managing hosts through the HMC. In earlier implementations, each HMC-managed host triggered the startup of a dedicated compute service on the PowerVC management server. This approach limited the number of manageable hosts and often required multiple PowerVC instances in larger environments.

CPNs address this limitation by allowing users to register standard Red Hat Enterprise Linux (RHEL) or SUSE Linux Enterprise Server VMs within the same subnet as the PowerVC management server. Once registered, these CPNs assume responsibility for starting compute services, offloading the management server and enabling support for a greater number of hosts.

This architecture eliminates the need for multiple PowerVC instances and introduces two distinct methods for scaling compute hosts, improving the efficiency and capacity of PowerVC deployments.

Figure 5-11 shows the addition of CPNs to a single-node PowerVC implementation to increase the number of HMC-managed hosts in the PowerVC instance.

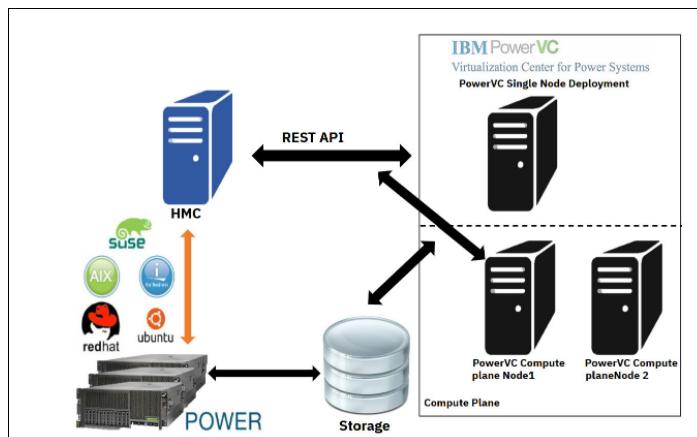


Figure 5-11 Single-node PowerVC with CPNs

Figure 5-11 on page 121 shows the addition of CPNs to a PowerVC HA multinode implementation to provide both scalability and HA.

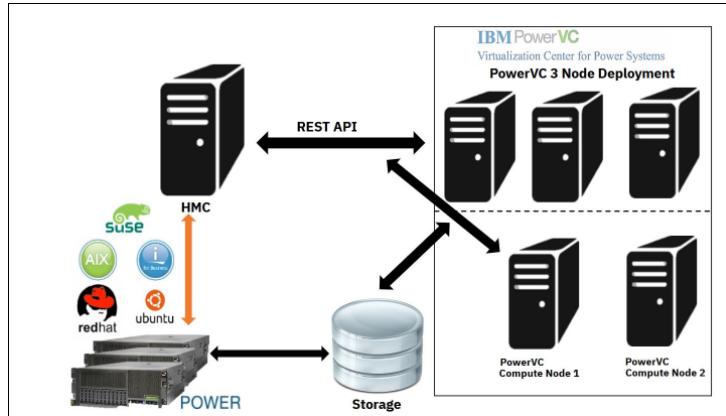


Figure 5-12 HA implementation of PowerVC with CPNs

5.3.2 PowerVC connectivity to managed systems

PowerVC provides two options for managing IBM Power servers: through an HMC or by using a NovaLink LPAR. When NovaLink is used, a dedicated partition must be created on each Power server to perform management functions similar to the ones that are provided by the HMC.

A combined solution is also supported, allowing both HMC and NovaLink to coexist. This hybrid approach offers flexibility in managing Power servers based on the specific requirements of the environment.

HMC management

The HMC is available as a physical or virtual appliance and serves as a central tool for managing IBM Power servers. It enables system administrators to configure and control one or more managed systems, including creating and managing LPARs and activating Capacity Upgrade on Demand. In addition to partitioning, the HMC supports essential service functions, such as detecting, consolidating, and transmitting system information to IBM Support. It acts as a central control point for Power hardware, helping ensure stability, efficient resource allocation, and streamlined service operations.

PowerVC uses REST API calls to connect to an HMC to create or modify LPARs on HMC-managed systems. If you typically use redundant HMCs to manage your hosts, you can continue this practice with PowerVC. Each host can be managed by multiple HMCs. The HMC through which the host is first registered becomes the primary HMC. If the host is also managed by a second HMC that is later added to PowerVC, that HMC is designated as the secondary HMC. If the primary HMC fails, PowerVC automatically fails over the host connection to the secondary HMC.

When using the compute node function, PowerVC can manage up to 45 HMC-managed hosts. Each host can support up to 1,000 VMs, with a maximum of 5,000 VMs across all hosts. Each HMC can manage up to 2,000 VMs.

NovaLink

PowerVM NovaLink is a software interface that is used for virtualization management. As of Version 2.3.0, PowerVM NovaLink installs on an RHEL partition running Versions 8.10, 9.2, or later. NovaLink enables highly scalable cloud management and deployment of critical enterprise workloads.

The PowerVM NovaLink installer supports rapid provisioning of large numbers of VMs. When you start from the System Management Services (SMS) interface of a Power server, the installer creates an RHEL PowerVM NovaLink partition and VIOS partitions, installs the operating systems, and deploys the NovaLink software. This process reduces installation time and supports repeatable deployments, which is beneficial in large scale-out environments.

NovaLink runs in a dedicated partition on each Power server that it manages. As shown in Figure 5-13, if you install the PowerVM NovaLink environment on a new managed system, the installer automatically creates the NovaLink partition. For systems that are already managed by an HMC, you must first create a Linux partition and then use the NovaLink installer to complete the installation.

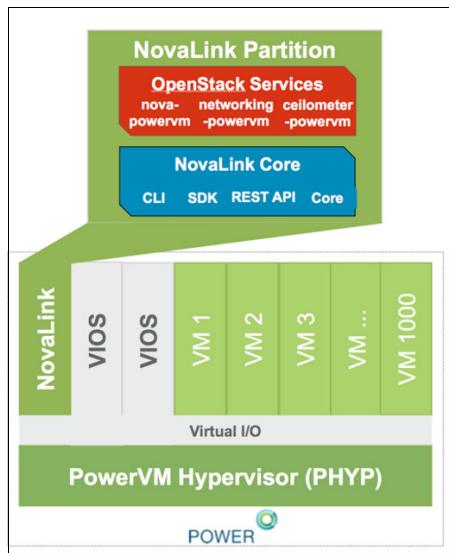


Figure 5-13 NovaLink architecture

PowerVM NovaLink reduces complexity and enhances the security of the server management infrastructure. It provides a dedicated server management network interface, which is secure by design and requires minimal user configuration. This interface facilitates secure communication between NovaLink and its managed VMs.

NovaLink offers the highest scalability for PowerVC managed environments. A single PowerVC instance can manage up to 60 NovaLink managed hosts. Across these hosts, PowerVC supports up to 10,000 VMs and 20,000 volumes. Each PowerVM host can support up to 1,000 VMs, including NovaLink, VIOS, and client workload partitions. These limits are based on the PowerVM platform firmware versions that were available at the time of the PowerVC 2.3.0 release.

NovaLink integrates with PowerVC and other OpenStack-based solutions to manage IBM Power servers in hybrid cloud environments.

For more information, see [NovaLink Whats New in PowerVM](#).

Combining HMC and NovaLink controlled systems

NovaLink managed systems and HMC-managed systems can coexist within your Power infrastructure, as shown in Figure 5-14. When a managed system is co-managed by the HMC and PowerVM NovaLink, you must set either PowerVM NovaLink or the HMC to controller mode. Certain system management, partition management, and Capacity on Demand (CoD) tasks can be performed only from the interface that is in controller mode.

For example, if PowerVM NovaLink is in controller mode, you can perform partition change operations only by using PowerVM NovaLink. To perform partition change operations by using the HMC, you must set the HMC to controller mode.

Firmware updates for a co-managed system can be performed only from the HMC. The HMC must be in controller mode to update the firmware. For more information, see [Managed systems co-management](#).

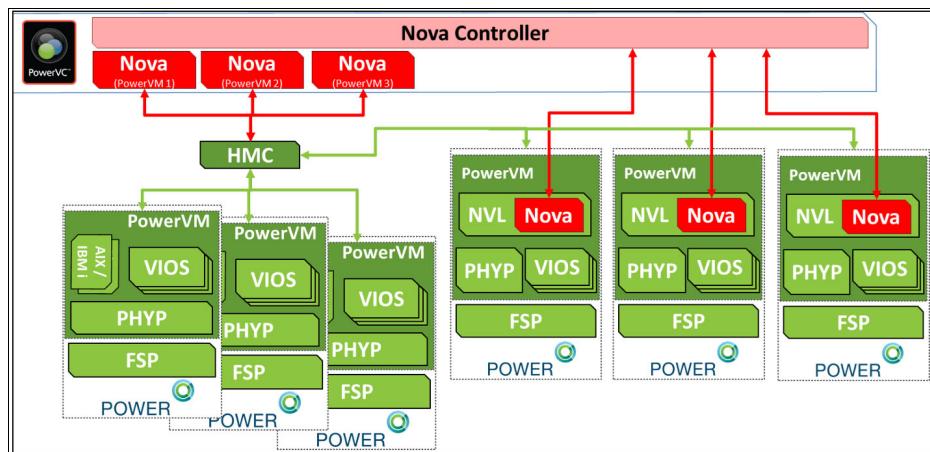


Figure 5-14 PowerVC with HMC and NovaLink partitions

When your environment includes both HMC-managed and NovaLink-managed hosts, the system supports a maximum of 3,000 VMs across all hosts. A maximum of 50 hosts is supported, of which up to 30 can be HMC-managed.

PowerVC OpenStack

PowerVC for Private Cloud 2.3.0 is built on OpenStack Caracal and supports Python 3.11. OpenStack Caracal 2024.1 introduces enhancements for AI and high-performance computing (HPC). It improves agility, performance, and security, and adds support for industry-leading hardware and software.

Note: For more information about the OpenStack Caracal release, see [2024.1 Caracal Release Highlights](#).

PowerVC streamlines the management of IBM Power servers by offering two primary connection methods: It can connect directly to the HMC, which acts as a proxy for accessing the managed systems, or it can connect directly to NovaLink partitions on those systems. This flexibility enables PowerVC to rapidly discover and interpret the PowerVM configuration of the managed systems, including the VIOSs and their connections to storage area network (SAN) fabrics and Ethernet networks. This immediate discovery reduces the need for extensive manual configuration, simplifying and accelerating the setup and management of virtualized environments.

The OpenStack software provides industry-standard interfaces that are released under the Apache License. PowerVC interfaces are a subset of OpenStack northbound APIs. Several interfaces were added or extended to enhance the capabilities that are associated with the IBM Power platform.

You can use several types of interfaces to build solutions on top of PowerVC:

- ▶ Supported OpenStack APIs: These APIs are a subset of the standard OpenStack APIs and can be used with PowerVC without modification.
- ▶ Extended OpenStack APIs: These APIs are based on OpenStack APIs but include more functions that are provided by PowerVC.
- ▶ PowerVC APIs: These APIs are exclusive to PowerVC and not part of OpenStack.

Supported OpenStack APIs can be accessed by using Ansible OpenStack modules. This access enables full automation of VM resource administration and deployment within the IBM Power managed system infrastructure.

Figure 5-15 shows the different automation options for configuring your IBM Power infrastructure.

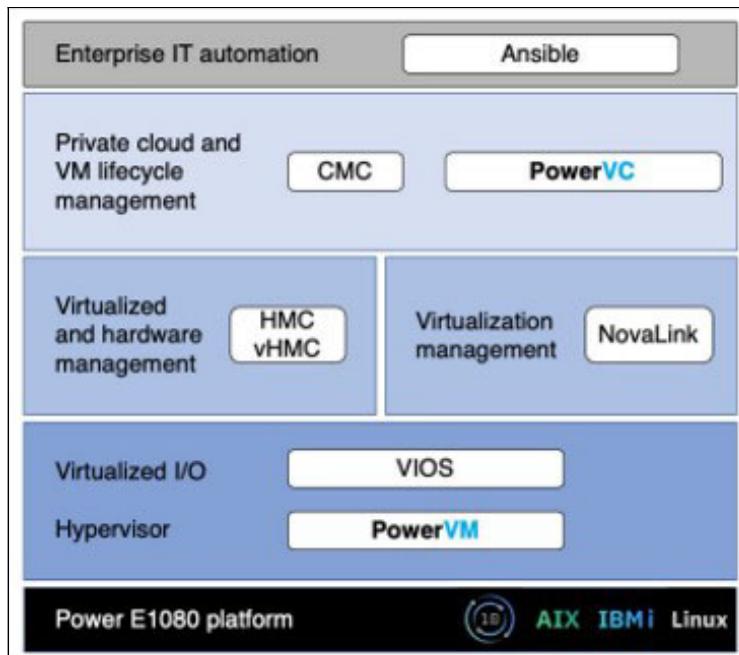


Figure 5-15 The Power Automation Stack

5.3.3 PowerVC automation

PowerVC is an advanced virtualization and cloud management solution. Built on OpenStack, it simplifies virtualization management and cloud deployments for IBM AIX, IBM i, and Linux VMs running on IBM Power servers. PowerVC is designed to build private cloud capabilities on Power servers and improve administrator productivity. It can also integrate with cloud environments through higher-level cloud orchestrators.

IBM Cloud Pak for AIOps is an AIOps platform that uses explainable AI and data from the IT operations (ITOps) toolchain to help you assess, diagnose, and resolve incidents across mission-critical workloads. It enables automation by using diverse datasets from hybrid environments, ranging from cloud to on-premises, and consolidates information across ITOps.

With IBM Cloud Pak for AIOps, you can access shared automation services to gain insight into how your processes run. You can visualize hotspots and bottlenecks, and use event detection to prioritize which issues to address first.

Infrastructure automation is available as an optional feature and is installed in a separate Red Hat OpenShift cluster from IBM Cloud Pak for AIOps. The Infrastructure Automation plug-in supports IBM PowerVC 1.4.4 or later.

In infrastructure automation, a manager is an external management environment that controls multiple resource types. One example is the PowerVC provider, which manages infrastructure, cloud, network, and storage resources.

Infrastructure automation includes the following components, which were previously available with IBM Cloud Pak for Multicloud Management:

- ▶ Infrastructure management (formerly IBM Red Hat CloudForms)

Infrastructure management provides the insight, control, and automation that are needed to address the complexities of virtual environments, which are more intricate than physical ones. This technology helps enterprises with existing virtual infrastructures improve visibility and control, and supports those beginning virtualization deployments in building and operating a managed virtual infrastructure.

- ▶ Managed services (formerly Terraform & Service Automation or IBM Cloud Automation Manager)

Managed services enable automated provisioning of infrastructure and VM applications across multiple cloud environments. Optional workflow orchestration is also supported.

Infrastructure automation enables ITOps and Site Reliability Engineers (SREs) to apply infrastructure as code (IaC) practices to increase IT velocity and shift operations earlier in the development lifecycle. It provides an integrated infrastructure automation capability for IT operators by using Terraform and ManageIQ.

If you have a license for IBM Cloud Pak for AIOps, you are entitled to install and use infrastructure automation.

Note: For more information about how to integrate PowerVC into IBM Cloud Pak for AIOps, see [IBM PowerVC Providers](#).

5.3.4 PowerVC setup tasks

Before you can use PowerVC, you must complete several configuration tasks, which are shown in Figure 5-16 on page 127.

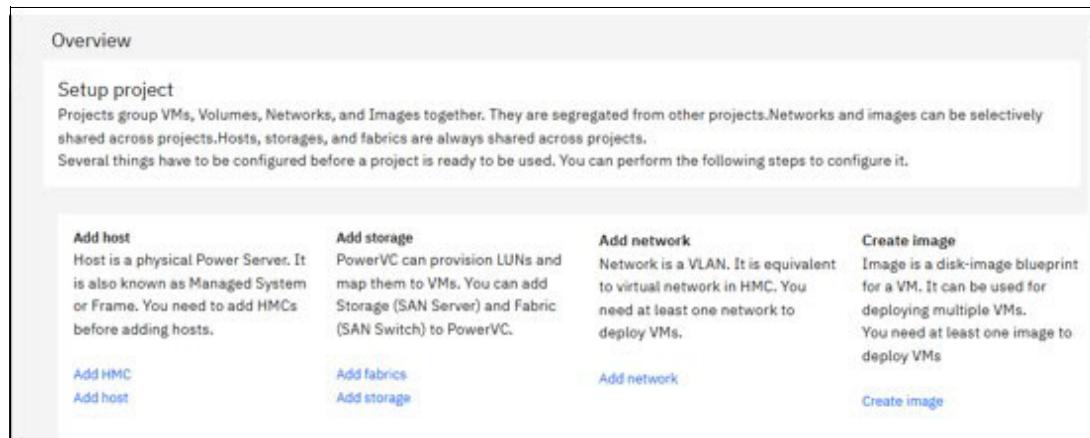


Figure 5-16 Initial window when logging in to a new installation

All configuration tasks are available in the PowerVC GUI.

Adding the HMC

Add the HMC that is attached to the Power managed servers, as shown in Figure 5-17.

	Name	Hostname or IP address	Hosts	Health
<input type="checkbox"/>	ibm.com	[Placeholder]	0	OK

Figure 5-17 Adding an HMC to PowerVC

The HMC connection and discovery process already has all the information of the connected IBM Power servers. PowerVC must obtain authorization for the HMC connection.

Adding a host

Once PowerVC is aware of an HMC, you can add a host definition to the PowerVC database, as shown in Figure 5-18. This discovery process is where PowerVC imports the managed system information into the PowerVC database.

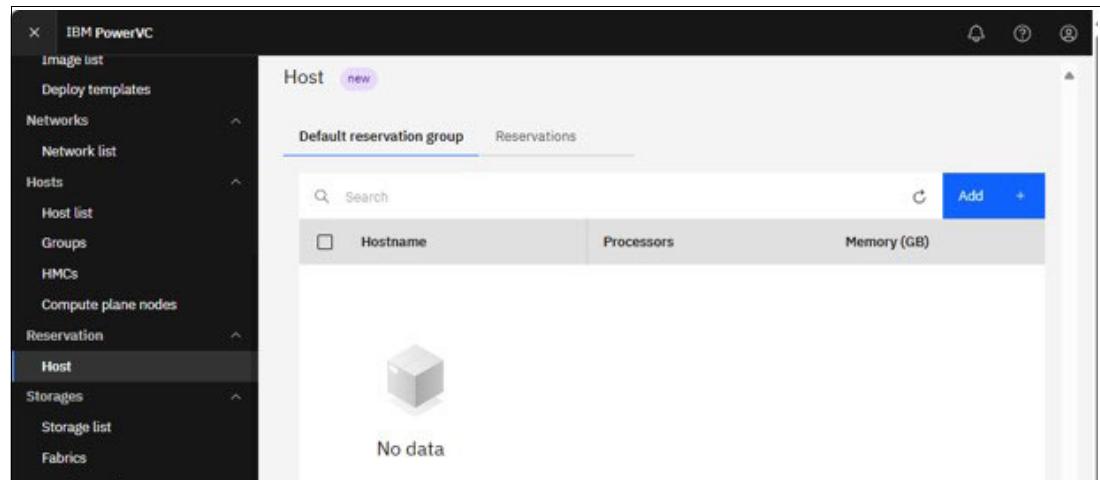


Figure 5-18 Adding a host

Adding a fabric

PowerVC supports a discovery process through the connected SAN fabric, where the VIOS Host Bus Adapters (HBAs) are zoned. This task is typically performed in collaboration with the storage administrator, who may provide a dedicated PowerVC user ID with administrative privileges to access the SAN storage and a defined storage pool for PowerVC deployments.

PowerVC also can discover existing storage and identify VMs that are using that storage.

Figure 5-19 shows adding a fabric through the GUI.

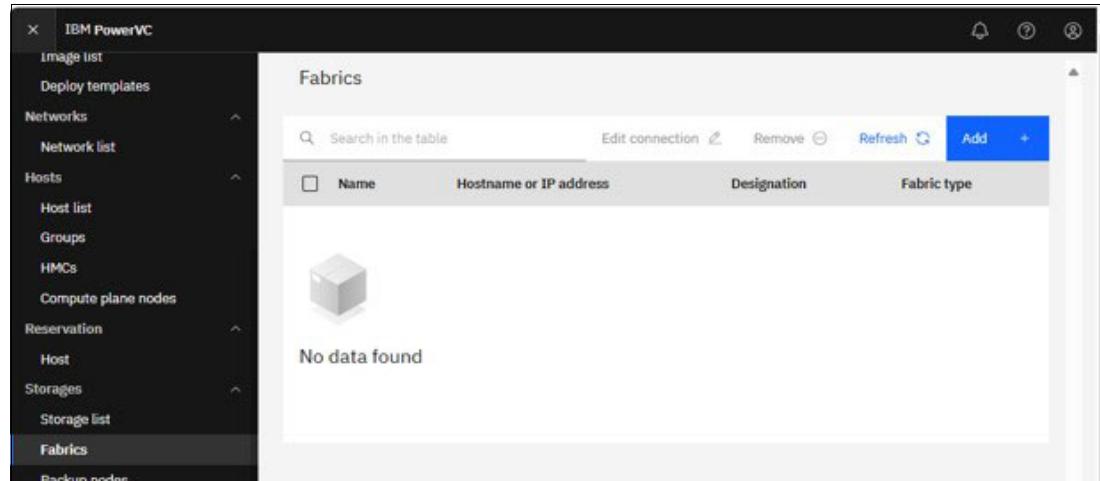


Figure 5-19 Adding a fabric

More details are required to determine the type of Fabric configuration that will be connected, as shown in Figure 5-20 on page 129.

Fabric list /

Add fabric

Display name *

Hostname or IP address *

User ID *

admin

Authentication type

Password SSH key

Password *

Zoning policy *

Initiator Initiator-target Initiator-vfc

Fabric type *

Brocade Cisco

Brocade fabric specific settings

Virtual fabric ID

Cisco fabric specific settings

Primary port *

22

VSAN *

Add

Figure 5-20 Adding fabric details

Adding storage

When access to the fabric is available, another discovery process is used to add a storage system to the SAN fabric, as shown in Figure 5-21.

IBM PowerVC

Host list

- Groups
- HMCs
- Compute plane nodes

Reservation

- Host
- Storages

Storage list

- Fabrics
- Backup nodes
- Boot volumes
- Data volumes
- Templates
- Connectivity groups

Storages

Search

Name	Hostname or IP address	Available space (GiB)	Total space (GiB)	State	Health	Volumes	Type
No data found							

Add

Figure 5-21 Adding a storage controller

PowerVC supports advanced storage functions, such as configuring Logical Unit Number (LUN) replication between storage systems. Implementing these functions is simpler when standard zoning practices are in place and might require coordination with the local storage administrator.

Help ensure that the SAN fabric zoning is compatible with LPM zoning requirements.

Adding a network

There is also a task for discovering any SEA VNets that are available on the VIOS, as shown in Figure 5-22.

Name	Type	VLAN ID	Gateway	Subnet mask	VM count	Free IPs
No data found						

Figure 5-22 Adding network connectivity

When PowerVC discovers a network connection that follows best practices and uses a dual-VIOS redundant configuration, it checks the VNet setup and indicates that redundancy is available.

Figure 5-23 on page 131 shows a configuration in which only an SEA-backed network is available for configuration. This view represents the Add Network window before any servers are added to PowerVC.

Networks /

Add network

Cancel
Add

Details

Name *	IP address type
<input type="text"/>	<input checked="" type="radio"/> DHCP <input checked="" type="radio"/> Static <input type="radio"/> Static IPv4 <input type="radio"/> IPv6 ⓘ
Network type	Subnet mask *
SEA - VLAN	Eg: 255.255.255.0
VIOS SEA - VLAN 1 - + <small>Specify a VLAN ID other than 1 to display Shared Ethernet Adapters for each host.</small>	Gateway *
<input type="checkbox"/> Jumbo frames (MTU = 9000)	Eg: 10.0.0.1
<input type="checkbox"/> Shared across projects	Primary DNS (recommended) Eg: 8.8.8.8
<input type="checkbox"/> Public network	Secondary DNS (recommended) Eg: 8.8.4.4
IP address range *	
Enter each IP / range in a new line. Eg : 10.0.0.4-10.0.0.30. Eg: 10.0.0.4-10.0.0.30	

Figure 5-23 Configuring the network connection

When a managed system is discovered, its capabilities are known, as shown in Figure 5-24.

Networks /

Add network

Cancel
Add

Details	SEA	SR-IOV																																			
The following SR-IOVs will be used.																																					
Physical network name *	Redundant physical network name																																				
<input type="text"/> default	— Select —																																				
<input type="checkbox"/> Add redundant																																					
Below table contains combined list of physical ports ⓘ																																					
Q. Search table: <table border="1"> <thead> <tr> <th>Physical port name</th> <th>Status</th> <th>Host</th> <th>Port speed</th> <th>Available capacity</th> <th>Available logical ports</th> <th>Physical network name</th> </tr> </thead> <tbody> <tr> <td>T1</td> <td>False</td> <td>[REDACTED]</td> <td>Unknown</td> <td>100</td> <td>0</td> <td>default</td> </tr> <tr> <td>T2</td> <td>False</td> <td>[REDACTED]</td> <td>Unknown</td> <td>100</td> <td>0</td> <td>default</td> </tr> <tr> <td>T3</td> <td>False</td> <td>[REDACTED]</td> <td>Unknown</td> <td>100</td> <td>0</td> <td>default</td> </tr> <tr> <td>T4</td> <td>False</td> <td>[REDACTED]</td> <td>Unknown</td> <td>100</td> <td>0</td> <td>default</td> </tr> </tbody> </table>			Physical port name	Status	Host	Port speed	Available capacity	Available logical ports	Physical network name	T1	False	[REDACTED]	Unknown	100	0	default	T2	False	[REDACTED]	Unknown	100	0	default	T3	False	[REDACTED]	Unknown	100	0	default	T4	False	[REDACTED]	Unknown	100	0	default
Physical port name	Status	Host	Port speed	Available capacity	Available logical ports	Physical network name																															
T1	False	[REDACTED]	Unknown	100	0	default																															
T2	False	[REDACTED]	Unknown	100	0	default																															
T3	False	[REDACTED]	Unknown	100	0	default																															
T4	False	[REDACTED]	Unknown	100	0	default																															

Figure 5-24 Adding network view with several systems discovered

Single Root Input/Output Virtualization

To prepare a Single Root Input/Output Virtualization (SR-IOV) adapter for sharing, use the HMC GUI. For more information, see [Modifying SR-IOV adapters](#).

SR-IOV supports the pass-through of Ethernet data from guest VMs directly to hardware. This approach improves performance by allowing data to pass from guest VMs to physical adapters with minimal processing, enabling a guest VM to achieve near wire-speed Ethernet performance. SR-IOV also supports more configuration options, such as quality of service (QoS), which enforces bandwidth allocations for guest VMs.

PowerVC can connect a VM that uses an SR-IOV adapter by using options 2, 3, or 4, depending on the specified redundancy levels, as shown in Figure 5-25.

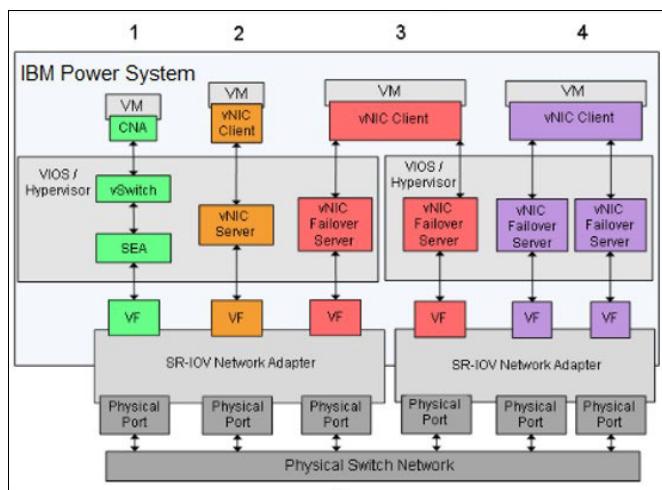


Figure 5-25 SR-IOV connectivity options

Note: For more information about using SR-IOV adapters with PowerVC, see [SR-IOV backed networks](#) and [Modifying SR-IOV adapters](#).

Adding an image

PowerVC enables you to capture and import images that can be deployed as VMs. An image consists of metadata and one or more binary files, one of which must be a bootable disk. To create a VM in PowerVC, you must deploy an image, as shown in Figure 5-26.

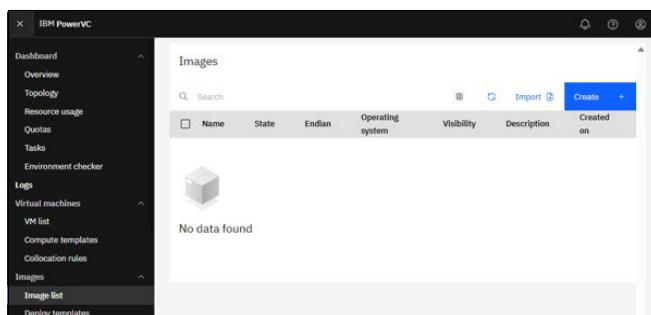


Figure 5-26 Managing images

Before a VM can be captured, it must meet specific requirements. If the VM is not properly prepared before capture, you might encounter issues when deploying the resulting image. For example, you might not be able to ping the VM that is created from the deployed image.

When capturing a VM, all volumes in its boot set are included in the generated image. If the VM is managed by PowerVC, the boot set includes all volumes that are marked as part of the boot set during management. If the VM is deployed from an image that is created within PowerVC, the boot set includes all volumes that are selected by the user during image creation. Unlike boot set volumes, data volumes are optional and can be selected by the user for inclusion in the captured image.

You can use cloud-init to prepare VMs for capture. Cloud-init is a technology that uses user input to configure the operating system and software on deployed VMs. It is widely used in OpenStack environments.

Note: For more information about working with images for PowerVC, see [Working with images](#).

Deploying a VM

After PowerVC is set up and the required resources are defined, you can deploy a VM. Using PowerVC, you can deploy a VM and make it available for login in approximately 5 minutes, depending on network and storage connectivity.

Figure 5-27 shows the list of VM deployments.

A screenshot of the IBM PowerVC web interface. The left sidebar has a dark theme with white text and includes links for Dashboard, Overview, Topology, Resource usage, Quotas, Tasks, Environment checker, Logs, Virtual machines, VM list (which is selected and highlighted in blue), Compute templates, Collocation rules, and Images. The main content area is titled 'Virtual machines' and contains a search bar and a table header with columns: Name, Host, IP, State, Health, Resources, Description, and VM owner. Below the table, there is a message 'No data found' and a small icon of a server rack.

Figure 5-27 VM list

5.3.5 VM deployment and automation scenarios

In an infrastructure without PowerVC, even if you use Ansible roles to automate HMC setup, including defining Fibre Channel (FC) adapter assignments, you are still responsible for several infrastructure tasks.

Figure 5-28 shows the remaining tasks, which are indicated in orange.

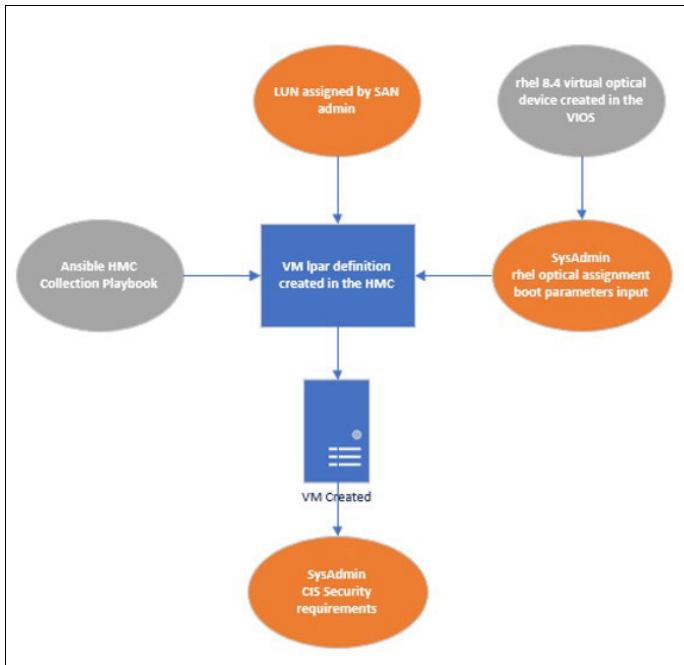


Figure 5-28 Remaining tasks

You can automate additional setup by using a kickstart file and **tftpboot** with a DHCP server. However, you must still coordinate with the SAN administrator, the security team, and the network team to provision IP addresses in DHCP, as shown in Figure 5-29.

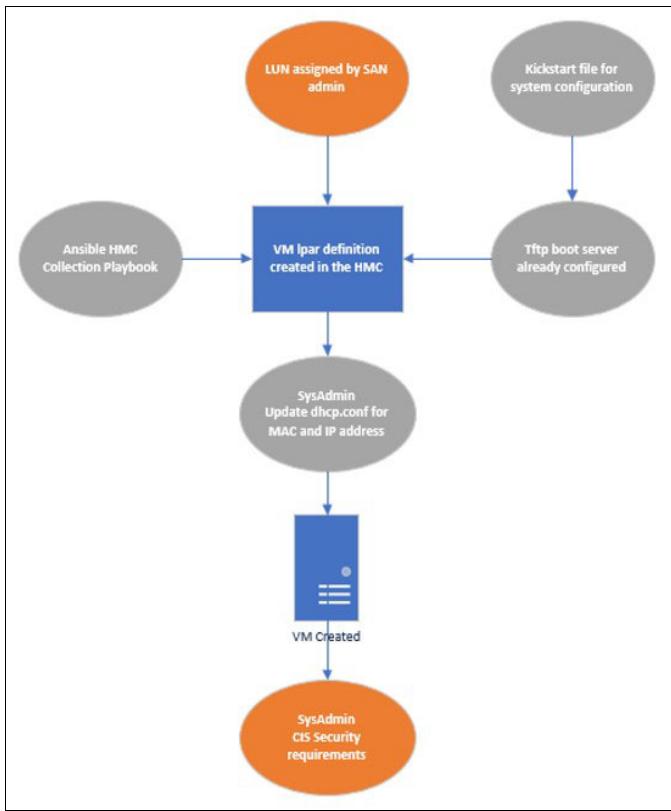


Figure 5-29 Deployment with tftpboot

After PowerVC is implemented to deploy a VM, you can use Ansible to provision the remaining components of a fully automated VM deployment solution, as shown in Figure 5-30.

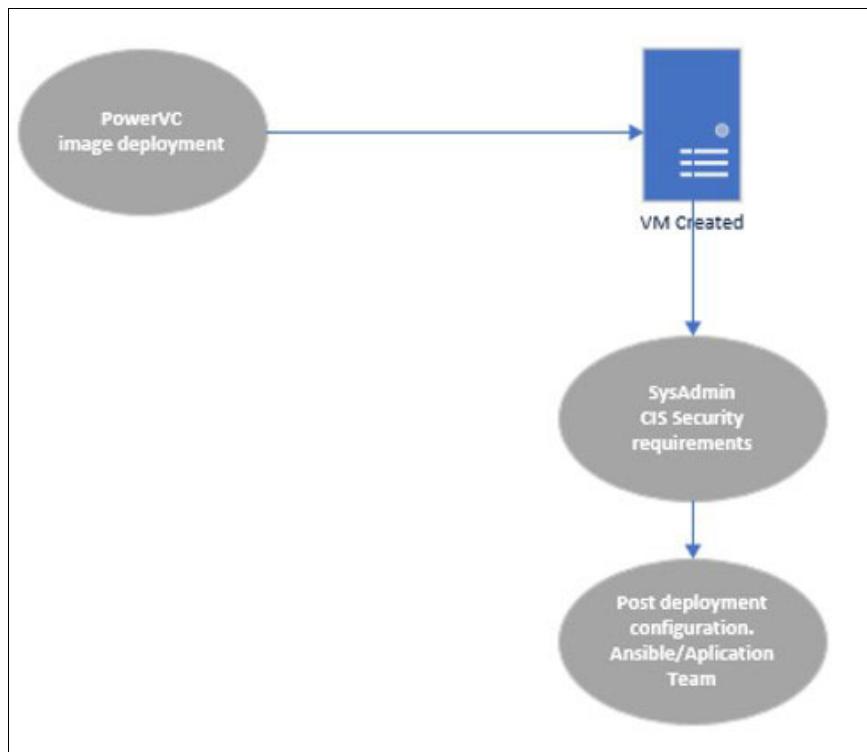


Figure 5-30 Deployment with PowerVC and Ansible

5.4 Ansible OpenStack Modules

PowerVC does not support all available [OpenStack commands](#). Only a subset of OpenStack commands are supported. Figure 5-31 shows some of the OpenStack commands.

Command	Description
openstack token issue	Issues a new token.
openstack token revoke	Revokes an existing token.
openstack port list	Lists ports.
openstack port delete	Deletes ports.

Figure 5-31 Sample OpenStack commands

However, more [OpenStack APIs](#) are available if you use Ansible OpenStack Modules. The `openstack.cloud` collection provides the APIs that are needed to interact with IBM PowerVC.

Figure 5-32 shows the Ansible OpenStack cloud server module website.

The screenshot displays the Ansible Community Documentation website. The left sidebar contains navigation links for Ansible Getting Started, Installation, Upgrade & Configuration, Using Ansible, Contributing to Ansible, and Ansible documentation style guide. The main content area is titled "openstack.cloud.server module – Create/Delete Compute Instances from OpenStack". It includes a "Note" section stating the module is part of the openstack.cloud collection (version 2.4.1), and instructions for installation using ansible-galaxy collection install openstack.cloud. Below this is a "Synopsis" section with a bullet point: "Create or Remove compute instances from OpenStack".

Figure 5-32 Ansible openstack.cloud.server module¹

Note: For more information about using Ansible for Automation on IBM Power servers, see *Using Ansible for Automation in IBM Power Environments*, SG24-8551.

5.4.1 An Ansible Execution Environment Image

Figure 5-33 shows the components of an Ansible Execution Environment Image (EEI). This approach provides the benefit of using an immutable image that can be pulled from the Ansible Hub and run on the Ansible Controller to access OpenStack APIs that are available in the PowerVC cloud.

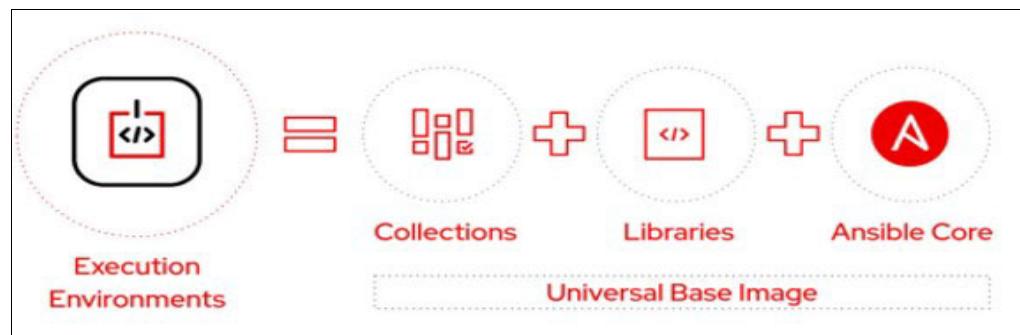


Figure 5-33 Components of an Execution Environment Image

The following sections demonstrate how to automate RHEL VM tasks by using Ansible Automation (ansible-navigator) on an IBM Power managed server, with PowerVC serving as the OpenStack cloud.

¹ Source: https://docs.ansible.com/ansible/latest/collections/openstack/cloud/server_module.html

5.4.2 Creating a collection

This section demonstrates how to create a collection for use with PowerVC running on IBM Power.

Complete the following steps:

1. The process begins with a set of existing roles, as shown in Example 5-2.

Example 5-2 Viewing the roles that are defined

```
[ansi01@controller modern]$ ls -ltr
total 44
-rw-rw-r--. 1 ansi01 ansi01 170 Feb 8 23:17 ansible-navigator.yml
drwxrwxr-x. 2 ansi01 ansi01 23 Feb 8 23:21 inventory
drwxrwxr-x. 3 ansi01 ansi01 17 Feb 8 23:36 group_vars
-rw-rw-r--. 1 ansi01 ansi01 792 Feb 9 15:02 site.yml
-rw-rw-r--. 1 ansi01 ansi01 151 Feb 9 15:04 ansible.cfg
drwxrwxr-x. 6 ansi01 ansi01 90 Feb 9 15:24 roles
-rw-rw-r--. 1 ansi01 ansi01 27756 Feb 9 15:55 ansible-navigator.log
-rw-rw-r--. 1 ansi01 ansi01 2540 Feb 9 15:55 ansible.log
[ansi01@controller modern]$ ls -ltr roles
total 0
drwxrwxr-x. 10 ansi01 ansi01 135 Nov 12 11:03 ocp_bootstrap
drwxrwxr-x. 10 ansi01 ansi01 135 Nov 12 11:03 os_add_volumes
drwxrwxr-x. 10 ansi01 ansi01 135 Nov 12 11:03 powervcvm
drwxrwxr-x. 10 ansi01 ansi01 135 Feb 9 15:38 ocp_nodes_create
[ansi01@controller modern]$
```

2. In this example, the powervcvm role calls the os_server module to create a VM in PowerVC by using the playbook at roles/powervcvm/tasks/powervc.yml. The Ansible Galaxy os_server module interacts directly with the PowerVC OpenStack cloud and the passes values that are shown in Example 5-3.

Example 5-3 Creating a VM by using PowerVC

```
- name: Create a VM in PowerVC
  os_server:
    state: present
    auth:
      auth_url: '{{ os_auth_url }}'
      username: '{{ os_username }}'
      password: '{{ os_password }}'
      project_name: '{{ os_project_name }}'
      user_domain_name: '{{ os_user_domain_name }}'
      project_domain_name: '{{ os_project_domain_name }}'
    timeout: 900
    validate_certs: no
    name: '{{ vm_name }}'
    image: '{{ powervc_rhel_image }}'
    flavor: '{{ worker_flavor }}'
    nics:
      - net-id: '{{ powervc_net_id }}'
    userdata: |
      {%- raw -%}#!/bin/bash
      service sshd restart
      {% endraw %}
```

```
register: vmout

- debug: var=vmout
  tags: [ never, debug ]
```

A playbook is also available for stopping and starting LPARs on IBM Power by using the HMC Ansible collection.

3. In Example 5-4, a server name is passed as a variable (`server_name`) and used to search for its details in the OpenStack PowerVC cloud namespace. The OpenStack details for the namespace are stored in a JSON dictionary object that is named `servers`.

Example 5-4 Getting the LPAR details

```
[ansi01@controller modern]$ cat roles/ocp_bootstrap/tasks/hmc_startup_nodes.yml
---

- name: get the lpar details from openstack
  set_fact:
    os_server_name: "{{ servers | community.general.json_query(jmespath_name) | replace('[','') | replace(']', '') }}"
    os_server_ip: "{{ servers | community.general.json_query(jmespath_ip) | replace('[','') | replace(']', '') }}"
    os_server_host: "{{ servers | community.general.json_query(jmespath_host) | replace('[','') | replace(']', '') | replace('9119MME_','Server-9119-MME-SN') }}"
  vars:
    jmespath_name: "servers[?name == '{{ server_name }}'].instance_name | [0]"
    jmespath_ip: "servers[?name == '{{ server_name }}']].access_ipv4 | [0]"
    jmespath_host: "servers[?name == '{{ server_name }}']].compute_host | [0]"
  register: lpar_details

- name: startup the lpar
  include_tasks: "stop_start_lpar.yml"
  vars:
    stop_start: 'poweron'
```

The `servers` dictionary was created in an earlier playbook at `roles/powervcvm/tasks/powervc.yml`. This dictionary is automatically inherited by subsequent Ansible playbooks and is referenced in Example 5-5.

Example 5-5 Retrieving the server list

```
- name: Retrieve list of all servers in this project
  os_server_info:
    auth:
      auth_url: '{{ os_auth_url }}'
      username: '{{ os_username }}'
      password: '{{ os_password }}'
      project_name: '{{ os_project_name }}'
      user_domain_name: '{{ os_user_domain_name }}'
      project_domain_name: "{{ os_project_domain_name }}"
    validate_certs: false
  register: servers
```

Inputs to the role are passed from Ansible as variables that are defined in the `group_vars` directory or through any other preferred input method.

4. Create a collection that is named `modern.powervc_ocp` for all roles in the `./collections` directory, as shown in Example 5-6.

Example 5-6 Creating a collection

```
[ansi01@controller modern]$ mkdir collections
[ansi01@controller modern]$ cd collections
[ansi01@controller collections]$ ansible-galaxy collection init modern.powervc_ocp
- Collection modern.powervc_ocp was created successfully
```

5. To include the roles in the collection, copy the roles into the roles location of the collection, as shown in Example 5-7.

Example 5-7 Copying the roles into the collection

```
[ansi01@controller collections]$ tree
.
???
modern
    ??? powervc_ocp
        ??? docs
        ??? galaxy.yml
        ??? meta
        ?   ??? runtime.yml
        ??? plugins
        ?   ??? README.md
        ??? README.md
        ??? roles

6 directories, 4 files
[ansi01@controller collections]$
[ansi01@controller collections]$ ls modern/powervc_ocp/roles/
[ansi01@controller collections]$
[ansi01@controller collections]$ cp -rp ../roles/* modern/powervc_ocp/roles/
[ansi01@controller collections]$ ls modern/powervc_ocp/roles/
ocp_bootstrap  ocp_nodes_create  os_add_volumes  powervcvm
[ansi01@controller collections]$
```

6. Update the `runtime.yml` collection with the Ansible version requirement, as shown in Example 5-8.

Example 5-8 Updating the collection

```
$ vim modern/powervc_ocp/meta/runtime.yml

add the following to the end of the file

requires_ansible: '>=2.9.10'
```

7. To import the collection into the Ansible Hub, update the `galaxy.yml` collection to define the galaxy collection prerequisites that are needed to run the playbooks in the collection, as shown in Example 5-9.

Example 5-9 Adding dependencies

```
$ vim modern/powervc_ocp/galaxy.yml
```

Add the following to the dependencies section.
Change this section

```
# range specifiers can be set and are separated by ','  
dependencies: {}
```

To

```
# range specifiers can be set and are separated by ','  
dependencies:  
    ansible.posix: '>=1.0.0'  
    openstack.cloud: '>=1.0.0'  
    community.general: '>=1.1.0'
```

5.4.3 Building the collection

To build the collection, complete the following steps:

1. Run the collection build, as shown in Example 5-10.

Example 5-10 Building a collection

```
[ansi01@controller powervc_ocp]$ ansible-galaxy collection build  
Created collection for modern.powervc_ocp at  
/home/ansi01/git-repos/modern/collections/modern/powervc_ocp/modern-powervc_ocp-1.  
0.0.tar.gz
```

2. Copy the collection tar.gz file to the location of your ansible.cfg where you have your Ansible Hub repository keys defined, as shown in Example 5-11.

Example 5-11 Copy collection file

```
[ansi01@controller powervc_ocp]$ cp modern-powervc_ocp-1.0.0.tar.gz ../../..  
[ansi01@controller powervc_ocp]$ cd ../../..
```

5.4.4 Uploading the collection to Ansible Hub

Upload the collection to Ansible Hub by completing the following steps:

1. Create the modern namespace on your Ansible Hub, as shown in Figure 5-34 on page 141.

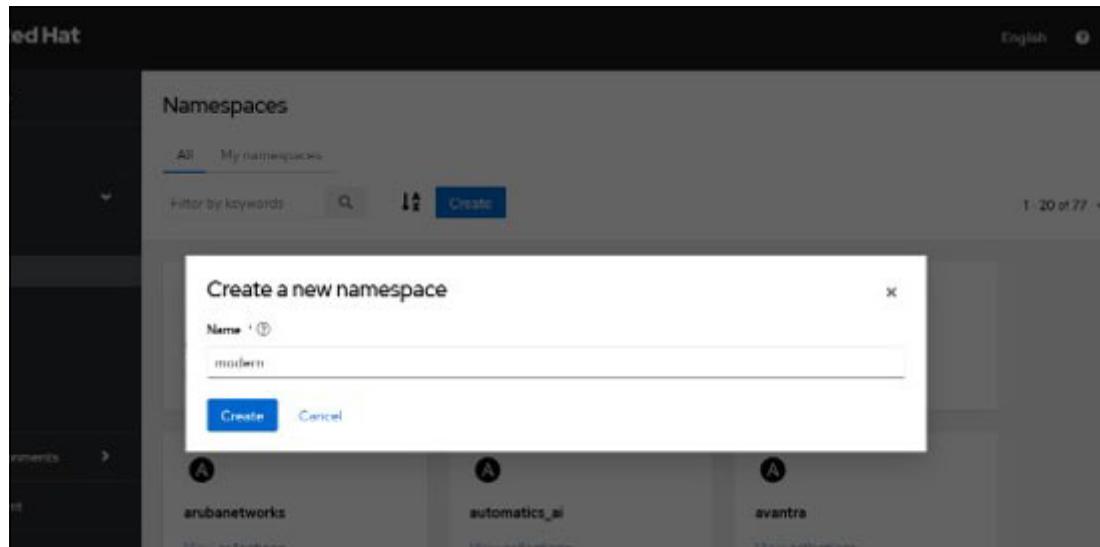


Figure 5-34 Creating a namespace

2. Ensure that you have the correct key for your Automation Hub in your ansible.cfg, as shown in Example 5-12.

Example 5-12 Validating the key

```
[defaults]
inventory = ./inventory/inventory
remote_user = ansi01
deprecation_warnings = false
log_path= ./ansible.log
forks = 20
collections_path = ./collections

[galaxy]
server_list = published, rh-certified, galaxy

[galaxy_server.published]
url=https://xx.xx.xx.xx/api/galaxy/
token=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

3. Upload the collection for use and availability, as shown in Example 5-13.

Example 5-13 Uploading the collection

```
$ ansible-galaxy collection publish modern-powervc_ocp-1.0.0.tar.gz --ignore-certs
```

- Approve the collection, as shown in Figure 5-35.

The screenshot shows the 'Approval dashboard' in the Red Hat Automation Hub. On the left, a sidebar menu includes 'Automation Hub', 'Search', 'Collections' (selected), 'Namespaces', 'Repositories', 'Remotes', 'API token', and 'Approval' (highlighted). The main area displays a table with one row:

Name	Collection	Version	Date created	Repository	Status
modern	powervc_ocp	1.0.0	24 minutes ago	staging	Needs review

A blue 'Approve' button is located at the bottom right of the table. Navigation controls like '1-1 of 1' and '1 of 1' are also visible.

Figure 5-35 Approving the collection

At this stage, a custom collection is available in the Ansible Automation Hub for use in an Ansible Execution Environment Image (EEI), as shown in Figure 5-36.

The screenshot shows the 'modern' namespace in the Red Hat Automation Hub. The sidebar menu is identical to Figure 5-35. The main area shows the 'Collections' tab selected. A collection named 'powervc_ocp' is listed:

- Icon:** Ansible logo (A)
- Name:** powervc_ocp
- Provided by:** modern
- Details:** 0 Modules, 4 Roles, 0 Plugins, 3 Dependencies
- Status:** Updated 26 minutes ago, v1.0.0 (published)
- Actions:** Upload new version

Figure 5-36 Custom collection

These custom roles are now available to any department in the organization as an Ansible Galaxy collection for use in their administrative tasks. The Ansible workstation or Ansible controller must include the prerequisite collections that are declared in the role dependencies, as shown in Example 5-14.

Example 5-14 Role dependencies

```
dependencies:
  ansible.posix: '>=1.0.0'
  openstack.cloud: '>=1.0.0'
  community.general: '>=1.1.0'
```

You should receive warning messages when uploading the collection confirming this availability, as shown in Example 5-15.

Example 5-15 Importing error messages

```
[ansi01@controller modern]$ ansible-galaxy collection publish modern-powervc_ocp-1.0.0.tar.gz
--ignore-certs
Publishing collection artifact '/home/ansi01/git-repos/modern/modern-powervc_ocp-1.0.0.tar.gz'
to published https://000.000.000.000/api/galaxy/
Collection published to the Galaxy server published https://000.000.000.000/api/galaxy/
Waiting until Galaxy import task
https://000.000.000.000/api/galaxy/v3/imports/collections/0194eb07-3c8f-750f-9844-2c07fa7b1c3d/
completed
[WARNING]: Galaxy import warning message: No changelog was found. Add a CHANGELOG.rst,
CHANGELOG.md, or changelogs/changelog.yaml file.
[WARNING]: Galaxy import warning message: roles/ocp_bootstrap/tasks/ocp_bootstrap.yml:3:6:
syntax-check[specific]: couldn't resolve
module/action 'os_server_info'. This often indicates a misspelling, missing collection, or an
incorrect module path.
[WARNING]: Galaxy import warning message: roles/ocp_nodes_create/tasks/os_create_nodes.yml:20:6:
syntax-check[specific]: couldn't resolve
module/action 'os_server_info'. This often indicates a misspelling, missing collection, or an
incorrect module path.
[WARNING]: Galaxy import warning message: roles/os_add_volumes/tasks/add_lun.yml:3:4:
syntax-check[specific]: couldn't resolve module/action
'os_server_volume'. This often indicates a misspelling, missing collection, or an incorrect
module path.
[WARNING]: Galaxy import warning message: roles/powervcvm/tasks/powervc.yml:3:4:
syntax-check[specific]: couldn't resolve module/action
'os_server'. This often indicates a misspelling, missing collection, or an incorrect module
path.
Collection successfully published and imported to the Galaxy server published
https://000.000.000.000/api/galaxy/
```

To prevent departments from maintaining workstations that might contain conflicting collections for various requirements, an Ansible EEI is created. This EEI includes the custom collection along with its required dependencies. The EEI can then run anywhere as a container.

5.4.5 Create the custom Execution Environment Image file

To create your EEI file, complete the following steps:

1. Log in to Podman for your Ansible Automation Hub.
2. Create a directory to work in by running the following command:
`mkdir ee-os-sdk`
3. Create the execution-environment.yml file for ansible-builder, as shown in Example 5-16.

Example 5-16 Creating the execution-environment yml file

```
---
version: 1
build_arg_defaults:
    EE_BASE_IMAGE: 'ansiblehub.xx.xx.xx/ee-supported-rhel8:latest'
```

```
EE_BUILDER_IMAGE: 'ansiblehub.xx.xx.xx/ansible-builder-rhel8:latest'
ansible_config: ansible.cfg
dependencies:
  galaxy: requirements.yml
  python: requirements.txt
  system: bindep.txt
```

Note: This example uses the execution-environment.yml file that is used with ansible-builder 1 for simplicity.

For more information about ansible-builder 3, see [Unlocking efficiency: Harnessing the capabilities of ansible-builder 3.0](#).

4. Create the requirements.yml file for the collections dependencies.

In Example 5-17, the newly created modern.powervc_ocp collection and the existing ibm.power_hmc collection are added.

Example 5-17 Creating the requirements.yml file

```
# cat requirements.yml
---
collections:
  - name: openstack.cloud
  - name: modern.powervc_ocp
  - name: ansible.posix
  - name: ansible.utils
  - name: ansible.netcommon
  - name: community.general
  - name: ibm.power_hmc
```

5. Create the requirements.txt file for Python, as shown in Example 5-18.

Example 5-18 Creating the requirements.txt file

```
# cat requirements.txt
openstackclient
openstacksdk
```

6. Create the bindeps.txt file for the OS, as shown in Example 5-19.

Example 5-19 Creating the bindeps.txt file

```
# cat bindeps.txt
libxml2-devel
libsasl-devel
python3-devel
gcc
python3-lxml
```

7. Copy the Red Hat subscription details of the local VM to two directories and create a rhsm-auths.tar file of the subscription details, as shown in Example 5-20.

Note: This step is no longer required in ansible-builder 3.

Example 5-20 Creating the .tar file

```
[ansi01@controller modern]$ mkdir rhsm-ca
[ansi01@controller modern]$ mkdir etc-pki-entitlement
[ansi01@controller modern]$ cp -rp /etc/rhsm/* rhsm-ca/
[ansi01@controller modern]$ cp -rp /etc/pki/entitlement/* etc-pki-entitlement/
[ansi01@controller modern]$ tree rhsm-ca
rhsm-ca
?? ca
? ??? redhat-entitlement-authority.pem
? ??? redhat-uep.pem
?? facts
? ??? insights-client.facts
?? logging.conf
?? pluginconf.d
?? rhsm.conf
?? syspurpose
??? syspurpose.json
??? valid_fields.json

4 directories, 7 files
[ansi01@controller modern]$ tree etc-pki-entitlement
etc-pki-entitlement
?? 5941199257546690002-key.pem
?? 5941199257546690002.pem

0 directories, 2 files
[ansi01@controller modern]$
[ansi01@controller modern]$ tar cvf rhsm-auths.tar rhsm-ca etc-pki-entitlement
rhsm-ca/
rhsm-ca/ca/
rhsm-ca/ca/redhat-entitlement-authority.pem
rhsm-ca/ca/redhat-uep.pem
rhsm-ca/facts/
rhsm-ca/facts/insights-client.facts
rhsm-ca/logging.conf
rhsm-ca/pluginconf.d/
rhsm-ca/rhsm.conf
rhsm-ca/syspurpose/
rhsm-ca/syspurpose/syspurpose.json
rhsm-ca/syspurpose/valid_fields.json
etc-pki-entitlement/
etc-pki-entitlement/5941199257546690002-key.pem
etc-pki-entitlement/5941199257546690002.pem
[ansi01@controller modern]$
```

- Verify that the files that are shown in Example 5-21 are present in your working directory.

Example 5-21 File list

```
[ansi01@controller ee-os-sdk]$ ls -ltr
total 260
-rw-rw-r--. 1 ansi01 ansi01    318 Feb  9 18:06 execution-environment.yml
-rw-rw-r--. 1 ansi01 ansi01    190 Feb  9 18:06 requirements.yml
-rw-rw-r--. 1 ansi01 ansi01     29 Feb  9 18:07 requirements.txt
-rw-rw-r--. 1 ansi01 ansi01     60 Feb  9 18:07 bindep.txt
-rw-rw-r--. 1 ansi01 ansi01   597 Feb  9 18:26 ansible.cfg
-rw-rw-r--. 1 ansi01 ansi01 245760 Feb  9 21:25 rhsm-auths.tar
```

- Build the context for the EEI, as shown in Example 5-22.

Example 5-22 Building the EEI context

```
[ansi01@controller ee-os-sdk]$ ansible-builder create
Complete! The build context can be found at:
/home/ansi01/git-repos/modern/ee-os-sdk/context
[ansi01@controller ee-os-sdk]$
[ansi01@controller ee-os-sdk]$ tree context
context
???
?? _build
?  ?? ansible.cfg
?  ?? bindep.txt
?  ?? requirements.txt
?  ?? requirements.yml
?  ?? scripts
?    ?? assemble
?    ?? check_ansible
?    ?? check_galaxy
?    ?? entrypoint
?    ?? install-from-bindep
?    ?? introspect.py
?? Containerfile

2 directories, 11 files
[ansi01@controller ee-os-sdk]$
```

- Copy the rhsm-auths.tar file in to the context directory, and extract it. Your tree should now look like the Example 5-23.

Example 5-23 Directory tree for rhsm-auths.tar

```
[ansi01@controller ee-os-sdk]$ cp rhsm-auths.tar context/
[ansi01@controller ee-os-sdk]$ cd context
[ansi01@controller context]$ tar xvf rhsm-auths.tar
[ansi01@controller context]$ rm rhsm-auths.tar
[ansi01@controller context]$ tree .
.
???
?? _build
?  ?? ansible.cfg
?  ?? bindep.txt
?  ?? requirements.txt
?  ?? requirements.yml
?  ?? scripts
```

```

?      ??? assemble
?      ??? check_ansible
?      ??? check_galaxy
?      ??? entrypoint
?      ??? install-from-bindep
?      ??? introspect.py
??? Containerfile
??? etc-pki-entitlement
?  ??? 5941199257546690002-key.pem
?  ??? 5941199257546690002.pem
??? rhsm-ca
    ??? ca
    ?  ??? redhat-entitlement-authority.pem
    ?  ??? redhat-uep.pem
??? facts
?  ??? insights-client.facts
??? logging.conf
??? pluginconf.d
??? rhsm.conf
??? syspurpose
    ??? syspurpose.json
    ??? valid_fields.json

```

8 directories, 20 files

11. Copy the text from Example 5-24 into the context/Containerfile, which enables the container to install the required rpm packages.

Example 5-24 Copying entitlements to context/Containerfile

```

# Base build stage
FROM $EE_BASE_IMAGE as base
USER root
ARG EE_BASE_IMAGE
ARG EE_BUILDER_IMAGE
ARG PYCMD
ARG PKGMGR_PRESERVE_CACHE
ARG ANSIBLE_GALAXY_CLI_COLLECTION_OPTS
ARG ANSIBLE_GALAXY_CLI_ROLE_OPTS

<==== copy the below HERE

# Copy entitlements
COPY ./etc-pki-entitlement /etc/pki/entitlement
# Copy subscription manager configurations
COPY ./rhsm-ca /etc/rhsm
COPY ./rhsm-ca/ca /etc/rhsm/ca

# clear the repo cache
RUN microdnf clean all

# Delete /etc/rhsm-host to use entitlements from the build container
RUN rpm -ivh
https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
RUN rm /etc/rhsm-host && \
    # Initialize /etc/microdnf.repos.d/redhat.repo

```

```

# See https://access.redhat.com/solutions/1443553
# microdnf repolist --disablerepo=* && \
microdnf install -y yum-utils && \
microdnf -y update && \
microdnf repolist && \
ls -l /etc/yum.repos.d && \
microdnf install -y --enablerepo=codeready-builder-for-rhel-8-x86_64-rpms
epel-release && \
microdnf install -y --enablerepo=openstack-17.1-for-rhel-8-x86_64-rpms
python3-openstacksdk && \
microdnf install -y --enablerepo=openstack-17.1-for-rhel-8-x86_64-rpms
python3-openstackclient && \
# Remove entitlements and Subscription Manager configs
rm -rf /etc/pki/entitlement && \
rm -rf /etc/rhsm

```

12. Change the builder image to minimal to make the image smaller and to ensure that microdnf works as expected, as shown in Example 5-25.

Example 5-25 Changing the image to minimal

```
ARG EE_BASE_IMAGE="ansiblehub.sbm.com.sa/ee-minimal-rhel8:latest"
```

If you need to, add an ignore certs for self-certified Ansible hubs for the lines that are shown in Example 5-26.

Example 5-26 Optional ignore certs

```
RUN ansible-galaxy role install --ignore-certs $ANSIBLE_GALAXY_CLI_ROLE_OPTS -r requirements.yml
--roles-path "/usr/share/ansible/roles"
RUN ANSIBLE_GALAXY_DISABLE_GPG_VERIFY=1 ansible-galaxy collection install --ignore-certs
$ANSIBLE_GALAXY_CLI_COLLECTION_OPTS -r requirements.yml --collections-path
"/usr/share/ansible/collections"
```

5.4.6 Building the Ansible Execution Environment Image

1. Use Podman to build the EEI. Example 5-27 shows the output (some text is truncated).

Example 5-27 Podman build

```
[ansi01@controller ee-os-sdk]$ Podman build -f context/Containerfile -t
ansiblehub.xx.xx.xx.xx/ee-modern-openstacksdk:19 context
[1/4] STEP 1/17: FROM ansiblehub.sbm.com.sa/ee-supported-rhel8:latest AS base
[1/4] STEP 2/17: USER root
--> Using cache ee86864116a460a65881029b8d58fa530b425551f2e469722e63adb551108e5a
--> ee86864116a4
[1/4] STEP 3/17: ARG EE_BASE_IMAGE
--> Using cache 06d3d71e4ce40539ef8ecb8fa694dbd073cd51e065d5295aa9b07e438acfcb
<< truncated >>
--> Using cache 06d3d71e4ce40539ef8ecb8fa694dbd073cd51e065d5295aa9b07e438acfcb
--> 06d3d71e4ce4
[1/4] STEP 4/17: ARG EE_BUILDER_IMAGE
--> Using cache 504ed2a0fb73b3cee0269da3e2f45d0b031d3b5caa424c0091fe58c1c96fb207
--> 504ed2a0fb73
[1/4] STEP 5/17: ARG PYCMD
```

```

--> Using cache 59d5892c581dbffb35c41bfae59062338470e3792df6d8d9a4ee4cb79bc7b929
--> 59d5892c581d
[1/4] STEP 6/17: ARG PKGMGR_PRESERVE_CACHE
--> Using cache d8b75224ccbc4407f0d689fabe6befb577a7b8d06ee1b7592afa417820ebe615
--> d8b75224ccbc
[1/4] STEP 7/17: ARG ANSIBLE_GALAXY_CLI_COLLECTION_OPTS
--> Using cache 6d81d88b752e738f8e6a0a63d75df5c7100300fb22ac95c4dfe870cacf1c43cb
--> 6d81d88b752e
[1/4] STEP 8/17: ARG ANSIBLE_GALAXY_CLI_ROLE_OPTS
--> Using cache b6928939ca458433eb10689c0f19cbaf3b8765e7701e8587cd2818c6a8b24929
--> b6928939ca45
[1/4] STEP 9/17: COPY ./etc-pki-entitlement /etc/pki/entitlement
--> 4e0a3b25e811
[1/4] STEP 10/17: COPY ./rhsm-ca /etc/rhsm
--> 6703339f6b5b
[1/4] STEP 11/17: COPY ./rhsm-ca/ca /etc/rhsm/ca
--> 6f6512c307fe
[1/4] STEP 12/17: RUN microdnf clean all
Complete.
--> 47488efb1972
[1/4] STEP 13/17: RUN rpm -ivh
https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
warning: /var/tmp/rpm-tmp.R28XbI: Header V4 RSA/SHA256 Signature, key ID 2f86d6a1:
NOKEY
Retrieving https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
Verifying... #####
Preparing... #####
Updating / installing...
epel-release-8-21.el8 #####
Many EPEL packages require the CodeReady Builder (CRB) repository.
It is recommended that you run /usr/bin/crb enable to enable the CRB repository.
--> 1e93bc6dce6b
[1/4] STEP 14/17: RUN rm /etc/rhsm-host && microdnf install -y yum-utils &&
microdnf -y update && microdnf repolist && ls -l /etc/yum.repos.d &&
microdnf install -y --enablerepo=codeready-builder-for-rhel-8-x86_64-rpms
epel-release && microdnf install -y
--enablerepo=openstack-17.1-for-rhel-8-x86_64-rpms python3-openstacksdk &&
microdnf install -y --enablerepo=openstack-17.1-for-rhel-8-x86_64-rpms
python3-openstackclient && rm -rf /etc/pki/entitlement && rm -rf /etc/rhsm
Downloading metadata...
Downloading metadata...
Downloading metadata...
Downloading metadata...

```

<< truncated >>

```

python-congressclient-2.0.1 python-dateutil-2.9.0.post0
python-designateclient-6.1.0 python-glanceclient-4.7.0 python-heatclient-4.1.0
python-ironic-inspector-client-5.2.0 python-ironicclient-5.10.0
python-keystoneclient-5.5.0 python-mistralclient-5.3.0 python-muranoclient-2.8.0
python-neutronclient-11.4.0 python-octaviaclient-3.9.0
python-openstackclient-7.2.1 python-saharaclient-4.2.0
python-searchlightclient-2.1.1 python-senlinclient-3.1.0 python-swiftclient-4.6.0
python-troveclient-8.7.0 python-vitrageclient-5.2.0 python-watcherclient-4.7.0
python-zaqarclient-2.10.0 python-zunclient-5.2.0 requests.exceptions-1.4.0

```

```

rfc3986-2.0.0 semantic-version-2.10.0 stevedore-5.4.0 textfsm-1.1.3 ttp-0.9.5
typing-extensions-4.12.2 tzdata-2025.1 ujson-5.10.0 warlock-2.0.1 wcwidth-0.2.13
websocket-client-1.8.0 wrapt-1.17.2 yaql-3.0.0
+ EXTRAS=
+ '[' -f /output/packages.txt ']'
++ wc -l
++ ls -1 '/output/wheels/*whl'
+ '[' 0 -gt 0 ']'
+ '[' '!' -z '' ']'
+ [[ '' != always ]]
+ /usr/bin/microdnf clean all
Complete.
+ rm -rf /var/cache/dnf /var/cache/yum
+ rm -rf /var/lib/dnf/history.sqlite /var/lib/dnf/history.sqlite-shm
/var/lib/dnf/history.sqlite-wal
+ rm -rf '/var/log/dnf.*' /var/log/hawkey.log
--> d390762f6eac
[4/4] STEP 11/12: RUN rm -rf /output
--> fa1559d888ed
[4/4] STEP 12/12: LABEL ansible-execution-environment=true
[4/4] COMMIT ansiblehub.sbm.com.sa/ee-openstacksdk:19
--> cb9ac6bc45f3
Successfully tagged ansiblehub.xx.xx.xx/ee-openstacksdk:19
cb9ac6bc45f3f14d3332ac11945b786c3432a0c50954f8a524587fc9134fa525
[ansi01@controller ee-os-sdk]$
```

2. Log in to your Ansible Automation Hub with Podman, as shown in Example 5-28.

Example 5-28 Logging in to Ansible Automation Hub

```
[ansi01@controller ee-os-sdk]$ Podman login ansiblehub.xx.xx.xx.xx
Username : admin
Password:
Login Succeeded!
```

3. Push the created EEI to your Ansible Automation Hub, as shown in Example 5-29.

Example 5-29 Pushing the file to Ansible Automation Hub

```
[ansi01@controller ee-os-sdk]$ Podman push ansiblehub.xx.xx.xx/ee-openstacksdk:19
Getting image source signatures
Copying blob fea8f7143030 done
Copying blob 5c70f3b118ad done
Copying blob 7a0d89c24bcf done
Copying blob 0c43d0e6782a done
Copying blob 0cc1efe62608 done
Copying blob f032c03593a1 done
Copying blob 1de299fb0d50 skipped: already exists
Copying blob c3df9143d763 done
Copying blob 6dca49abef01 done
Copying blob dc9b17c0a14e skipped: already exists
Copying blob 9bf490c38d15 done
Copying blob 87e3ab05d9a4 skipped: already exists
Copying blob 0754bf7bf972 done
Copying blob 18a7217d9b08 done
Copying blob bf757458f9cd done
Copying blob 83b46ea877e6 done
```

```
Copying config cb9ac6bc45 done |  
Writing manifest to image destination  
[ansi01@controller ee-os-sdk]$
```

You can now use this EEI to run any playbooks that use Ansible OpenStack and IBM HMC modules.

5.4.7 Populating the environment variables

The EEI includes all the collections that are required to run a playbook. You need only the previously created playbooks and an `ansible.cfg` file. The playbooks include a disk attachment that is named by the environment variable `new_vol`.

Complete the following steps:

1. Create a disk in PowerVC with any name, for example, `modern`, as shown in Figure 5-37.

Data volumes						
<input type="checkbox"/>	Name	Size (GiB)	State	Health	VMs	Storage template
<input type="checkbox"/>	modern	10	Available	OK	0	V7K Power8 STRG base template V7K Power8 STRG
Items per page: 100		1–1 of 1 items				
						1 ▾ of 1 page ▶

Figure 5-37 Data volume list

2. Update the playbook variables in `group_vars/all/infra.yml`.

The `site.yml` file creates one VM that is named `vm01` and three worker nodes, and uses the RHEL image from PowerVC, as shown in Example 5-30.

Example 5-30 The `group_vars/all/infra.yml` file

```
---  
  
# HMC access  
hmc_username: 'xxxxx'  
hmc_password: 'xxxxxx'  
hmc_hostname: 'xxxxxxxx'  
  
# VMs to Create  
vm01_hostname: "vm01"  
  
workers_list:  
    - worker_name: "worker-0"  
    - worker_name: "worker-1"  
    - worker_name: "worker-2"  
  
# name of the disk created in PowerVC to be attached to vm01  
new_vol: "modern"
```

~
~

3. The other environment file that must be populated is group_vars/all/os_powervc.yml. Enter the cloud information for the PowerVC project that you plan to use. Also, copy the PowerVC certificate from /etc/pki/tls/certs/powervc.crt to the workstation.

Example 5-31 shows these actions.

Example 5-31 The group_vars/all/os_powervc.yml file

```
os_identity_api_version: 3
os_auth_url: "https://xx.xx.xx.xx:5000/v3"
os_cert: "/etc/pki/tls/certs/powervc.crt"
os_region_name: "RegionOne"
os_project_domain_name: "Default"
os_project_name: 'modern'
os_tenant_name: "{{ os_project_name }}"
os_user_domain_name: "Default"
os_username: 'xxxxx'
os_password: 'xxxxx'
os_compute_api_version: 2.46
os_network_api_version: 3
os_image_api_version: 2
os_volume_api_version: 3
powervc_rhel_image: '5c262de6-91c8-4bec-bf39-edbeef2918c'
powervc_coreos_image: '06799bd3-d2e8-4968-86ca-8f1945d37f34'
worker_flavor: '52121faf-0fca-4eab-881b-779e622377de'
master_flavor: '3a0736cb-74aa-4bf0-a5da-f09cb6679f54'
auto_ip: 'yes'
powervc_net_id: '5529e29d-2127-4a78-b3dd-ebd42baa89b4'
```

4. Add the UUID of the PowerVC images that you will use.
5. Add the UUID of PowerVC flavors that you will use.

5.4.8 Using the EEI to run an Ansible playbook

Complete the following steps:

1. Build vm01 and the worker nodes by running the previously created site.yml file, as shown in Example 5-32.

Example 5-32 Running the Ansible playbook

```
[ansi01@controller modern]$ ansible-navigator --eei
ansiblehub.xx.xx.xx/ee-modern-openstacksdk:1.19 run site.yml -m stdout
```

```
PLAY [Deploy some VMs]
*****
TASK [modern.powervc_ocp.powervcvm : Create a VM in PowerVC]
*****
changed: [localhost]
```

```

TASK [modern.powervc_ocp.powervcvm : Retrieve list of all servers in this project]
*****
ok: [localhost]

TASK [modern.powervc_ocp.powervcvm : debug]
*****
ok: [localhost] => {
    "msg": "vm name is vm01"
}

TASK [modern.powervc_ocp.powervcvm : Get the created server]
*****
ok: [localhost] => {
    "msg": [
        {
            "access_ipv4": "xx.xx.xx.xx",
            "id": "29c2bfd0-f36c-4f3d-9785-bbb93e16339a",
            "name": "vm01",
            "status": "ACTIVE"
        }
    ]
}

TASK [modern.powervc_ocp.powervcvm : get the server name and IP Fact]
*****
ok: [localhost]

TASK [modern.powervc_ocp.powervcvm : update eei /etc/hosts for xx.xx.xx.xx vm01]
*****
changed: [localhost]

TASK [modern.powervc_ocp.powervcvm : Pause for 2 minutes to allow the interface to
be up] *****
Pausing for 120 seconds
(Ctrl+C then 'C' = continue early, Ctrl+C then 'A' = abort)

```

2. Update the /etc/hosts file of the running container image with the new VM details to update Ansible facts, if needed.

Figure 5-38 shows the creation of the VM in PowerVC.

Name	Host	IP	State	Health	Resources	Description	VM owner	Expiration date	Running task
vm01	[REDACTED]	[REDACTED]	Building	Pending	4 vCPU / 0 GB / 0 PU				Building

Figure 5-38 Building vm01

Figure 5-39 shows the building of the worker nodes.

Name	Host	IP	State	Health	Resources	Description	VM owner	Expiration date	Running task
vm01	Server-9119-MME- xxxxxxxxxx	xxxxxxxxxx	Active	Warning	4 vCPU /16 GB /1 PU				
worker-0	Server-9119-MME- xxxxxxxxxx	xxxxxxxxxx	Building	Pending	4 vCPU /0 GB /0 PU				Building

Figure 5-39 Building the workers

- PowerVC typically allows you to create multiple VMs with the same name by appending an OpenStack UUID to each instance. If you run this playbook a second time, the VMs are not created again. The system checks OpenStack for the UUIDs of the existing VMs, as shown in Example 5-33.

Example 5-33 Re-running the playbook

```
[ansi01@controller modern]$ ansible-navigator --eei
ansiblehub.xx.xx.xx.xx/ee-modern-openstacksdk:1.19 run site.yml -m stdout

PLAY [Deploy some VMs]
*****
TASK [modern.powervc_ocp.powervcvm : Create a VM in PowerVC]
*****changed: [localhost]

TASK [modern.powervc_ocp.powervcvm : Retrieve list of all servers in this project]
*****ok: [localhost]

TASK [modern.powervc_ocp.powervcvm : debug]
*****ok: [localhost] => {
    "msg": "vm name is vm01"
}

TASK [modern.powervc_ocp.powervcvm : Get the created server]
*****ok: [localhost] => {
    "msg": [
        {
            "access_ipv4": "xx.xx.xx.xx",
            "id": "29c2bfd0-f36c-4f3d-9785-bbb93e16339a",
            "name": "vm01",
            "status": "ACTIVE"
        }
    ]
}

TASK [modern.powervc_ocp.powervcvm : get the server name and IP Fact]
*****ok: [localhost]

TASK [modern.powervc_ocp.powervcvm : update eei /etc/hosts for xx.xx.xx.xx vm01]
*****changed: [localhost]
```

```
TASK [modern.powervc_ocp.powervcvm : Pause for 2 minutes to allow the interface to  
be up] ****  
Pausing for 120 seconds  
(Ctrl+C then 'C' = continue early, Ctrl+C then 'A' = abort)
```

4. You can access extensive information about the VM in JSON format from the PowerVC cloud. This data can be used to automate additional tasks in your environment, as shown in Example 5-34.

Example 5-34 Additional automations with information from PowerVC

```
TASK [modern.powervc_ocp.ocp_nodes_create : debug]  
*****  
ok: [localhost] => {  
    "vmout": {  
        "changed": true,  
        "failed": false,  
        "server": {  
            "access_ipv4": "",  
            "access_ipv6": "",  
            "addresses": {  
                "VLAN-130": [  
                    {  
                        "OS-EXT-IPS-MAC:mac_addr": "fa:26:12:23:65:20",  
                        "OS-EXT-IPS:type": "fixed",  
                        "addr": "xx.xx.xx.xx",  
                        "version": 4  
                    }  
                ]  
            },  
            "admin_password": null,  
            "attached_volumes": [  
                {  
                    "attachment_id": null,  
                    "bdm_id": null,  
                    "delete_on_termination": true,  
                    "device": null,  
                    "id": "6fa4131f-8fa0-47db-9653-4353873aa312",  
                    "location": null,  
                    "name": null,  
                    "tag": null,  
                    "volume_id": null  
                }  
            ]  
        }  
    }  
}
```

All servers are now created, as shown in Figure 5-40.

Virtual machines										
<input type="checkbox"/>	Name	Host	IP	State	Health	Resources	Description	VM owner	Expiration date	Running task
<input type="checkbox"/>	vm01	Server-1	192.168.1.101	Active	Green	4 vCPU / 16 GB / 1 PU				
<input type="checkbox"/>	worker-0	Server-2	192.168.1.102	Active	Yellow	4 vCPU / 16 GB / 1 PU				
<input type="checkbox"/>	worker-1	Server-3	192.168.1.103	Active	Yellow	4 vCPU / 16 GB / 1 PU				
<input type="checkbox"/>	worker-2	Server-4	192.168.1.104	Active	Yellow	4 vCPU / 16 GB / 1 PU				

Figure 5-40 All servers created

5.5 Automation tools

Ansible and Terraform are both powerful IaC tools that serve different purposes and complement each other effectively. Terraform is designed to provision and manage infrastructure resources across various cloud providers, such as IBM Cloud, Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP), and on-premises environments. It focuses on the infrastructure layer, defining and creating the underlying components.

Ansible is designed for configuration management, application deployment, and task automation. It focuses on configuring and managing software and settings on existing infrastructure.

Ansible and Terraform work together in a pipeline. Terraform creates the necessary infrastructure resources, such as VMs, networks, and databases. After the infrastructure is provisioned, Ansible configures the software and settings on those resources. Together, they provide a comprehensive solution for managing both the infrastructure and the software running on it. This automation reduces manual effort and errors.

By helping ensure consistent deployment and configuration, this approach simplifies scaling infrastructure and applications. Each tool focuses on its specific strengths, resulting in a more organized and efficient workflow. Combining their capabilities enables the creation of a powerful and streamlined infrastructure automation pipeline.

5.5.1 Ansible

Ansible is an open-source, cross-platform tool for resource provisioning automation. DevOps professionals use it for continuous deployment or delivery (CD) of software code by using an IaC approach. The Ansible Automation Platform evolved to deliver sophisticated automation solutions for operators, administrators, and IT decision-makers across various technical disciplines. It is an enterprise automation solution that is built on flourishing open-source software.

Ansible operates on several UNIX-like platforms and can manage systems across UNIX and Microsoft architectures. It uses a descriptive language to define system settings.

Due to its broad acceptance, open-source design, and wide support for many devices and platforms, Ansible is becoming a dominant tool in the market. However, it is common to use other automation tools alongside Ansible to support more complex automation workflows. For example, many organizations use Ansible with Terraform to automate infrastructure provisioning.

Ansible architecture

As shown in Figure 5-41, the Ansible architecture consists of an Ansible Controller and one or more Ansible client hosts. The controller runs automation tasks and houses Ansible collections, which contain modules, plug-ins, and roles defining the actions Ansible can perform on client nodes.

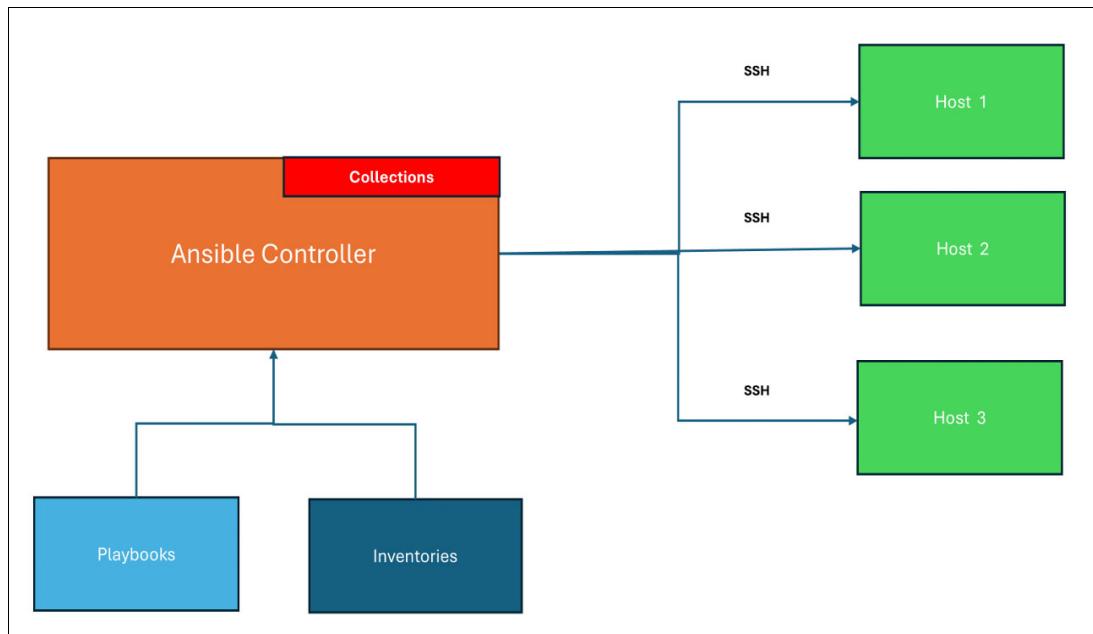


Figure 5-41 Simplified Ansible architecture

Playbooks

At the core of Ansible Automation are Ansible playbooks, which are YAML files that define sequences of tasks to run on remote hosts. These tasks can include installing packages, configuring services, or copying files. Playbooks enable IT teams to automate infrastructure provisioning, configuration management, application deployment, and other operational tasks.

Why choose Ansible

Ansible offers numerous benefits for IT professionals who aim to improve efficiency, scalability, and consistency in their infrastructure. Here are some key advantages:

- ▶ **Versatility:** Ansible supports a wide range of devices and can scale to meet growing automation needs.
- ▶ **Agentless architecture:** Ansible manages devices by using Secure Shell (SSH), which eliminates the need for agents on target systems.
- ▶ **Flexibility:** Ansible can be used for both simple CLI tasks and complex workflows that are defined in playbooks.
- ▶ **Extensive module library:** Ansible provides a rich collection of modules for managing systems, cloud infrastructures, and OpenStack environments.

- ▶ Declarative approach: With Ansible's declarative syntax, you define the state of a system, and Ansible performs the necessary steps to achieve it.
- ▶ Ease of learning: The YAML-based syntax and minimal learning curve make Ansible accessible to IT professionals at all levels.

Ansible is a powerful automation tool that helps organizations improve efficiency, scalability, and reliability in their IT infrastructure. By using Ansible playbooks, IT teams can streamline routine tasks, automate complex workflows, and help ensure consistent configurations across environments.

Options for implementing Ansible

As you implement Ansible for IT management, it is important to select the appropriate product and support level to meet your organization's requirements. This section describes several available options.

Ansible Community

The community versions of Ansible primarily include the following ones:

- ▶ **Ansible Core**

Ansible Core is the foundation of the Ansible ecosystem. It provides the core automation engine and includes essential functions for configuration management, application deployment, and task automation. Ansible Core also includes modules, plug-ins, and the CLI that is required to run playbooks and manage configurations.

- ▶ **AWX**

AWX is the upstream, open-source project that serves as the community version of Red Hat Ansible Tower. It provides a web-based UI, a Representational State Transfer (REST) API, and a task engine for managing Ansible automation at scale. AWX supports role-based access control (RBAC), job scheduling, graphical inventory management, and other features that help users manage and scale automation efforts.

- ▶ **Ansible Collections**

Ansible Collections are pre-packaged sets of modules, roles, and plug-ins that are created and shared by the community. Collections extend Ansible functions and are often maintained by community contributors or specific organizations. They can be downloaded from Ansible Galaxy, a central hub for sharing and discovering Ansible content.

- ▶ **Ansible Galaxy**

Ansible Galaxy is a community-driven repository for sharing and discovering Ansible roles and collections. It provides a searchable platform where users can find reusable content to simplify automation tasks. Roles and collections from Galaxy can be integrated directly into automation workflows.

These community versions are suitable for individual users, small teams, and development environments. However, they do not include the formal support and advanced features that are available in the Red Hat Ansible Automation Platform.

Ansible Automation Platform

Ansible Automation Platform is a subscription-based enterprise solution that integrates more than 20 community projects into a fully supported automation platform. It provides curated, certified, and validated Ansible Collections and roles from partners such as IBM, Juniper, Cisco, and major public cloud providers.

Here are the key considerations for choosing Ansible Automation Platform:

- ▶ Support level: Ansible Automation Platform offers enterprise-grade support, including service-level agreements (SLAs) for security, compatibility, and upgrades. Community options typically offer limited support.
- ▶ Features: The platform includes features beyond Ansible Core, such as a web interface and integration with other enterprise tools.
- ▶ Cost: Ansible Automation Platform is a subscription-based product. Community versions are available at no cost.
- ▶ Scale and complexity: For large organizations with complex automation requirements, Ansible Automation Platform might be the preferred choice due to its advanced features and enterprise support.

By evaluating these factors, you can choose the Ansible solution that best aligns with your organization's goals, budget, and support needs.

For more information about implementing automation with Ansible in an IBM Power environment, see *Using Ansible for Automation in IBM Power Environments*, SG24-8551.

5.5.2 Terraform

Terraform is an open-source tool that is developed by HashiCorp. It is written in the Go programming language and compiles into an executable program that is named Terraform. Terraform is an IaC tool that enables you to build, change, and create versions of cloud and on-premises resources safely and efficiently.

Terraform provides a mechanism to access any API from any cloud provider to manage infrastructure as a service (IaaS). Figure 5-42 illustrates the process of calling the API. The configuration files define which APIs to call. These files represent the code that is referenced in the IaC approach.



Figure 5-42 Terraform functions

Managing any infrastructure

Terraform connects to various providers, including Ansible, to manage infrastructure. You can browse the Terraform Registry to discover available providers. The Terraform Provider for Ansible offers a more straightforward and robust method for running Ansible automation from Terraform, compared to using the local-exec provisioner.²

² Source: <https://developer.hashicorp.com/terraform/language/resources/provisioners/local-exec>

Figure 5-43 shows the Ansible provider entry in the Terraform Registry.

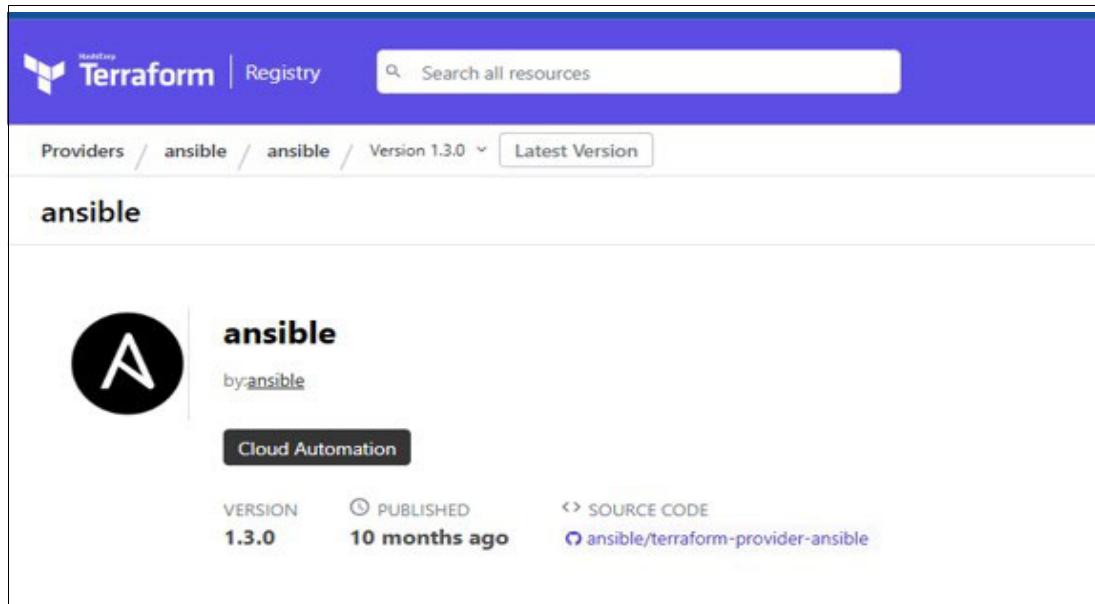


Figure 5-43 Terraform Ansible provider

The prerequisites for using the Ansible provider are as follows:

1. Install Go.

For installation instructions, see the [official installation guide](#).

2. Install Terraform.

Install the ppc64le version for operation on IBM Power. Installation instructions are found on the [GitHub registry](#).

3. Install Ansible.

To install Ansible, see the [Ansible official installation guide](#).

Tracking your infrastructure

Terraform generates a plan and prompts you for approval before modifying your infrastructure. The state of your infrastructure is stored in a file that is named `terraform.tfstate`. This file can be stored in Git, GitLab, or HCP Terraform to enable creating versions, and encryption and secure sharing with your team. It serves as the single source of truth for your environment.

Automating changes

Terraform configuration files are declarative, describing the wanted end state of infrastructure, so it is simple to automate them by using tools such as Ansible.

Standardizing configurations

Terraform provides standardization through modules. A module consists of a collection of `.tf` and `.tf.json` files that are stored together in a directory. Modules are the primary method for packaging and reusing resource configurations in Terraform.

Collaborating

Because Terraform configurations can be distributed as files and version-controlled by using platforms such as Git, GitHub, and HCP Terraform, they are ideal tools for sharing and collaboration.

5.5.3 Ansible Automation and Terraform

Terraform and Ansible provide the full Hybrid Cloud Lifecycle management to build and manage applications. Figure 5-44 shows how Ansible and Terraform work together.



Figure 5-44 Terraform and Ansible working together

The core Terraform workflow consists of three stages:

- ▶ Write
Define the infrastructure resources, which may span multiple cloud providers and services.
- ▶ Plan
Terraform generates a plan that describes the actions that it takes to create, update, or destroy infrastructure based on the state and your configuration.
- ▶ Apply
Upon approval, Terraform performs the proposed operations in the correct order, respecting resource dependencies.

Figure 5-45 shows this process.

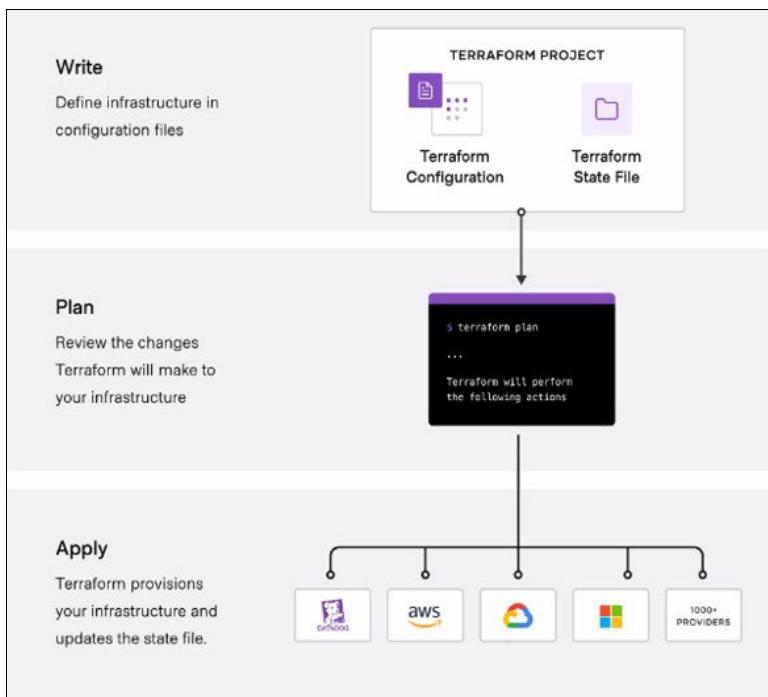


Figure 5-45 Terraform workflow

5.5.4 Terraform plan

In this use case, an x86 VM was used to run Terraform. This VM can also serve as the Ansible Controller.

Installing Terraform

Go to the Terraform site and follow the instructions for your platform, as shown by the following steps:

1. For RHEL on x86, add the Terraform repository and install Terraform by using dnf, as shown in Figure 5-46.

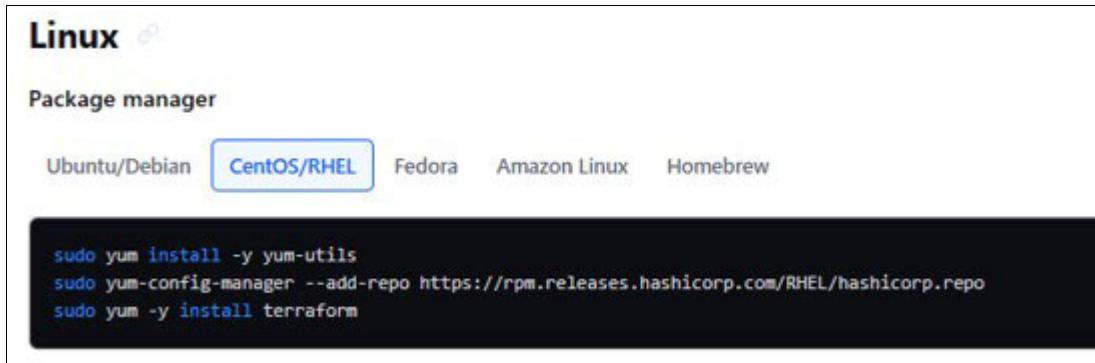


Figure 5-46 Installing Terraform

2. Create a working directory, as shown in Example 5-35 on page 163.

Example 5-35 Creating a working directory

```
# mkdir terra-ocp  
# cd terra-ocp
```

3. Define the provider in your Terraform configuration. In this use case, you use PowerVC OpenStack, so specify the OpenStack provider.

Figure 5-47 shows the OpenStack provider entry in the Terraform Registry.

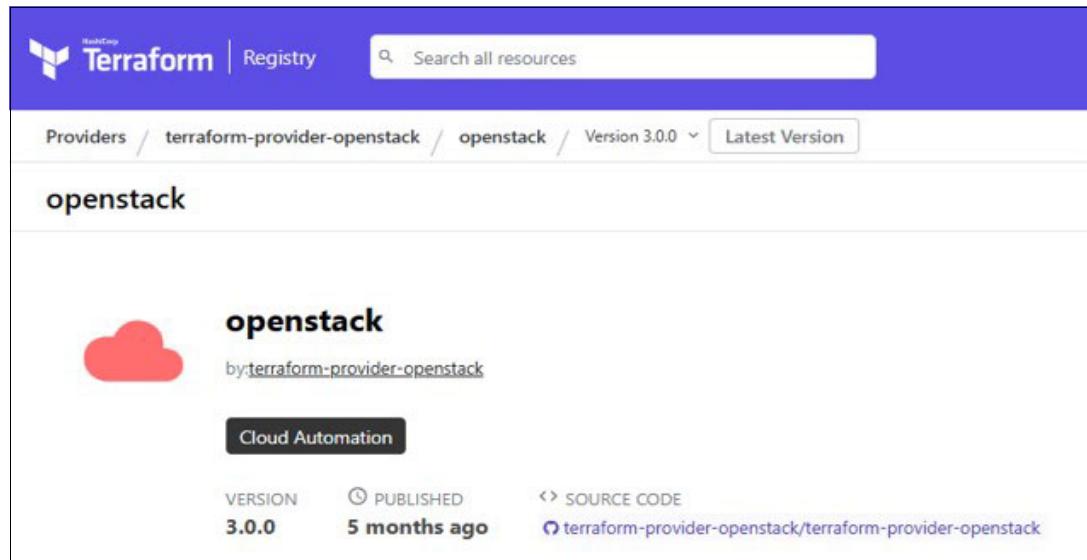


Figure 5-47 Terraform registry entry for OpenStack³

4. Create a file that is named providers.tf. In this file, specify the latest versions of the OpenStack, Random, and Terraform from the Terraform website.

Figure 5-48 shows the OpenStack provider configuration that is used in this example.

```
terraform {  
    required_providers {  
        openstack = {  
            source  = "terraform-provider-openstack/openstack"  
            version = "~> 3.0.0"  
        }  
        random = {  
            source  = "hashicorp/random"  
            version = "~> 3.7.1"  
        }  
    }  
    required_version = ">= 1.11.0"  
}  
  
provider "openstack" {  
    user_name      = var.os_username  
    password      = var.os_password  
    tenant_name   = var.os_project_name  
    domain_name   = var.os_user_domain_name  
    auth_url      = var.os_auth_url  
    insecure      = var.os_validate_certs  
}  
  
resource "random_id" "label" {  
    count         = var.cluster_id == "" ? 1 : 0  
    byte_length   = "2" # Since we use the hex, the word lenght would double  
    prefix        = "${var.cluster_id_prefix}-"  
}
```

Figure 5-48 Terraform file

³ Source: <https://registry.terraform.io/providers/terraform-provider-openstack/openstack/latest>

5. In this use case, Random is used to generate a unique ID for labeling LPAR names and hostnames.

OpenStack requires several variables, which are the same OpenStack cloud values that were used earlier in the Ansible Execution Environment playbooks. In Terraform, variables are referenced by using the var prefix and must be declared either in a variables file or passed on the CLI.

For more information, see [Providing Terraform with that Ansible Magic](#).

6. Create a file that is named variables.tf. In this file, define all the variables that are used in the Terraform plan.

Figure 5-49 shows an example of how these variables are declared.

```
variable "os_network_api_version" {
    type = string
    description = "The PowerVC openstack neutron version"
    default     = "3"
}

variable "os_image_api_version" {
    type = string
    description = "The PowerVC openstack glance version"
    default     = "2"
}

variable "os_storage_api_version" {
    type = string
    description = "The PowerVC openstack cinder version"
    default     = "3"
}

variable "powervc_rhel_image_uuid" {
    type = string
    description = "A base RedHat image to use for RHEL servers"
}

variable "powervc_coreos_image_uuid" {
    type = string
    description = "The coreos image to use for this deployment"
}

variable "worker_flavor" {
    type = string
    description = "The compute template created for workers in PowerVC"
}

variable "control_flavor" {
    type = string
    description = "The compute template created for control nodes in PowerVC"
}
```

Figure 5-49 The variables.tf file

You can define the type of each variable, add a description, and specify a default value. If no default value is provided, Terraform prompts you to enter a value when the plan runs. Alternatively, you can supply values by using a variables file that is passed on the CLI.

The vars.tfvars file is used to import values for variables that do not have default values that are defined in variables.tf. Figure 5-50 on page 165 shows examples of these variable definitions.

```
### PowerVC Details
os_auth_url = "https://:5000/v3"
os_username = ""
os_password = ""
os_project_name = ""
os_cert = "powervc.crt"
os_validate_certs = "false"
os_compute_api_version = "2.46"
os_network_api_version = "3"
os_image_api_version = "2"
os_storage_api_version = "3"
powervc_rhel_image_uuid = ""
powervc_coreos_image_uuid = ""
worker_flavor = ""
control_flavor = ""
storage_flavor = ""
rhel_flavor = ""
powervc_net_id = ""
powervc_net_name = ""
```

Figure 5-50 vars.tfvars

The variables that are declared so far are referred to as input variables.

7. Declare the output variables in a file that is named output.tf. Output variables capture values that are produced at run time that you want to retain or reference later. In this example, the output variable is cluster_id, as shown in Figure 5-51.

```
output "cluster_id" {
  value = local.cluster_id
}
```

Figure 5-51 Output variable

For more information about Terraform variables, see [Variables and Outputs](#).

8. Terraform automatically reads all files in the working directory with a .tf suffix. Based on the file type, it determines how to process each file.

In this example, a file that is named ocp.tf is created to define the actions of the Terraform plan. This file contains the resource definitions and logic for the deployment. Two methods are used to define these actions:

- a. Locals

A local value assigns a name to an expression, allowing you to reference the name multiple times within a module instead of repeating the expression. In this use case, the local value that is defined in the ocp.tf file is responsible for generating the cluster_id. This point in the configuration is where the cluster_id is created.

Example 5-36 shows creating a local value.

Example 5-36 Creating a local value

```
locals {  
    # Generates cluster_id as combination of cluster_id_prefix + (random_id or  
    user-defined cluster_id)  
    cluster_id = var.cluster_id == "" ? random_id.label[0].hex :  
    (var.cluster_id_prefix == "" ? var.cluster_id :  
    "${var.cluster_id_prefix}-${var.cluster_id}")  
}
```

b. Modules

Modules are containers for multiple resources that are used together. A module consists of a collection of .tf or .tf.json files that are stored in a directory. Modules are the primary method for packaging and reusing resource configurations in Terraform.

In this use case, each module is organized in its own directory within a parent directory that is named modules, as shown in Example 5-37.

Example 5-37 Defining modules

```
module "bastion" {  
    source = "./modules/1_bastion"  
  
    cluster_domain      = var.cluster_domain  
    cluster_id          = local.cluster_id  
    bastion             = var.bastion  
    powervc_net_id     = var.powervc_net_id  
    powervc_net_name   = var.powervc_net_name  
    rhel_flavor         = var.rhel_flavor  
    powervc_rhel_image_uuid = var.powervc_rhel_image_uuid  
}
```

Several variables are redeclared in these module declarations. These variables are passed to the directory that defines the plan for creating the resource, which in this case is a VM. The variables that are provided are required to create a VM in PowerVC. A similar module is declared for each VM to be created, as shown in Example 5-38.

Example 5-38 List of modules

```
[ansi01@controller terraform-ocp-upi]$ grep module ocp.tf  
module "bastion" {  
    source = "./modules/1_bastion"  
module "haproxy_1" {  
    source = "./modules/1_a_haproxy"  
module "haproxy_2" {  
    source = "./modules/1_b_haproxy"  
module "dns" {  
    source = "./modules/1_dns"  
[ansi01@controller  
terraform-ocp-upi]$
```

9. Resources are the most important elements in the Terraform language. Each resource block defines one or more infrastructure objects, such as VNets, compute instances, or higher-level components like DNS records.

In this example, each module directory includes a resource block for its corresponding VM module definition. Figure 5-52 shows the bastion resource block, which creates the bastion VM. This VM is an LPAR with a hostname of \${var.cluster_id}-bastion.

```
resource "openstack_compute_instance_v2" "bastion" {
  name = "${var.cluster_id}-bastion"
  image_id = var.powervc_rhel_image_uuid
  flavor_id = var.rhel_flavor

  network {
    uuid = var.powervc_net_id
    name = var.powervc_net_name
  }
}
```

Figure 5-52 Bastion VM resource block

Other variables for the PowerVC OpenStack cloud are already defined in the provider configuration. Therefore, only the name of the LPAR, the image that it uses, and the CPU and memory resources (based on the flavor that is defined in PowerVC) must be specified. For the network, either the `uuid` or the `name` value is required. Both are shown for demonstration purposes and can also serve as a verification method for the instance.

The contents of the bastion module directory include a subset of the variables that are required for this module, a repeated provider declaration, and the resource definition in the `bastion.tf` file, as shown in Example 5-39.

Example 5-39 Bastion node directory contents

```
[ansi01@controller 1_bastion]$ pwd
/home/ansi01/git-repos/terraform-ocp-upi/modules/1_bastion
[ansi01@controller 1_bastion]$ ls -ltr
total 12
-rw-rw-r--. 1 ansi01 ansi01 247 Mar  3 18:09 bastion.tf
-rw-rw-r--. 1 ansi01 ansi01 200 Mar  4 00:42 variables.tf
-rw-rw-r--. 1 ansi01 ansi01 425 Mar  4 01:06 providers.tf
[ansi01@controller 1_bastion]$
```

10. You can now run the command to generate a plan and review the expected changes. If the plan runs successfully, Terraform is properly configured and ready to apply the planned changes.

Figure 5-53 shows the successful output from the plan.

```
[ansi01@controller terraform-ocp-upi]$ terraform plan -var-file var.tfvars

Terraform used the selected providers to generate the following execution plan. Resources: 0 to add, 0 to change, 0 to destroy.

Terraform will perform the following actions:

# random_id.label[0] will be created
+ resource "random_id" "label" {
    + b64_std      = (known after apply)
    + b64_url      = (known after apply)
    + byte_length  = 2
    + dec          = (known after apply)
    + hex          = (known after apply)
    + id           = (known after apply)
    + prefix        = "-"

}

# module.bastion.openstack_compute_instance_v2.bastion will be created
+ resource "openstack_compute_instance_v2" "bastion" {
    + access_ip_v4      = (known after apply)
    + access_ip_v6      = (known after apply)
    + all_metadata      = (known after apply)
    + all_tags          = (known after apply)
    + availability_zone = (known after apply)
    + created           = (known after apply)
    + flavor_id         = (known after apply)
    + flavor_name       = (known after apply)
    + force_delete      = false
    + id                = (known after apply)
    + image_id          = (known after apply)
    + image_name        = (known after apply)
    + name              = (known after apply)
    + power_state       = "active"
    + region            = (known after apply)
    + security_groups   = (known after apply)
    + stop_before_destroy = false
    + updated           = (known after apply)

    + network {
        + access_network = false
        + fixed_ip_v4    = (known after apply)
        + fixed_ip_v6    = (known after apply)
        + mac             = (known after apply)
        + name            = (known after apply)
        + port            = (known after apply)
        + uuid            = (known after apply)
    }
}
```

Figure 5-53 Terraform plan output

11. When ready, run the same command with the **apply** option. Terraform prompts for confirmation, as shown in Example 5-40.

Example 5-40 Confirmation prompt

Plan: 5 to add, 0 to change, 1 to destroy.

Changes to Outputs:

~ cluster_id = "-d3b8" -> (known after apply)

Do you want to perform these actions?
Terraform will perform the actions that are described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

Figure 5-54 shows the created VMs in PowerVC.

Name	Host	IP	State	Health	Resources	Description	VM owner	Expiration date	Running task
ocpitest-712e-bastion			Building	Pending	1 vCPU /0 GB /0 PU	ocpitest-712e-bastion			
ocpitest-712e-dns			Building	Pending	1 vCPU /0 GB /0 PU	ocpitest-712e-dns			Building
ocpitest-712e-haproxy-1			Building	Pending	1 vCPU /0 GB /0 PU	ocpitest-712e-haproxy-1			
ocpitest-712e-haproxy-2			Building	Pending	1 vCPU /0 GB /0 PU	ocpitest-712e-haproxy-2			Building
ocpitest-712e-haproxy-3			Building	Pending	1 vCPU /0 GB /0 PU	ocpitest-712e-haproxy-3			

Figure 5-54 PowerVC GUI showing the created VMs

5.5.5 Configuring and customizing Terraform defined resources

Your VMs are installed, but no applications are installed and the VMs are not customized. To avoid having to manually set up each LPAR, further automation is required.

At this stage, the VMs are installed, but no applications are deployed, and the VMs are not customized. To avoid manually configuring each LPAR, more automation is required.

Using Terraform provisioners

You can complete the customization of the VM by using the local-exec or remote-exec provisioners in Terraform. However, it is a best practice to minimize the use of provisioners. Configuration management tools such as Ansible or cloud-init are often better suited for complex configuration tasks. In addition, provisioners can make Terraform configurations less predictable and more difficult to maintain.

Terraform supports two types of provisioners:

- ▶ The local-exec provisioner: This provisioner invokes a local executable file after a resource is created. It runs on the machine running Terraform, and not on the target resource.
- ▶ The remote-exec provisioner: This provisioner runs a script on the remote resource after it is created. It can be used to run configuration management tools or perform tasks such as bootstrapping into a cluster. The remote-exec provisioner requires a connection and supports both SSH and WinRM.

Provisioners are best suited for simple, one-time tasks that cannot be accomplished through other means. For more information about using provisioners in Terraform, see the official [HashiCorp documentation](#).

Using the Ansible provider for Terraform

Although Terraform provisioners can be used to customize VMs, it is a best practice to use Ansible for final customization. This section describes how to use the Ansible provider for Terraform.

Ansible provider prerequisites

To use the Ansible provider for Terraform, ensure that the following prerequisites are in place:

- ▶ Install Go, as described at [Download and install](#).
- ▶ Install Terraform, as described at [Install Terraform](#).
- ▶ Install Ansible, as described at [Installing Ansible](#).

Installing the prerequisites

Complete the following steps:

1. In this environment, Terraform and Ansible are already installed. Therefore, only Go must be installed. After completing the installation, verify it as shown in Example 5-41.

Example 5-41 Installing Go

```
[ansi01@controller terraform-ocp-upi]$ go version  
go version go1.24.0 linux/amd64
```

2. Download and make `terraform-provider-ansible`. Put the provider into the same `registry.terraform.io` for the project where you have OpenStack, as shown in Example 5-42.

Example 5-42 Downloading `terraform-provider-ansible`

```
[ansi01@controller .terraform]$ pwd  
/home/ansi01/git-repos/terraform-ocp-upi/.terraform  
[ansi01@controller .terraform]$ ls providers/registry.terraform.io/  
hashicorp  terraform-provider-ansible  terraform-provider-openstack
```

Note: For more information about how to download and make `terraform-ansible-provider`, see [GitHub](#).

The provider directory includes example playbooks. However, if you already have existing plays and collections that are integrated into an Ansible EEI, converting them into simplified playbooks might require extra effort. This process can also expose code on the controller.

5.5.6 Ansible Terraform Collection

The Ansible Collection `cloud.terraform` can help integrate existing Ansible EEIs into your Red Hat Ansible Automation Platform environment. In this scenario, Ansible is used to call Terraform, as shown in Figure 5-55 on page 171.

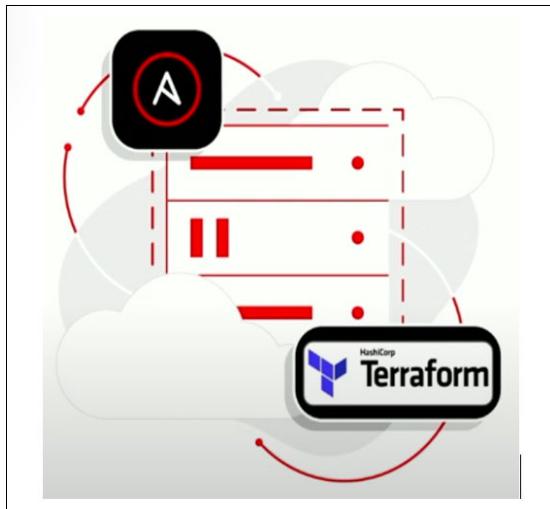


Figure 5-55 Ansible and Terraform working together

Figure 5-56 shows the collection.

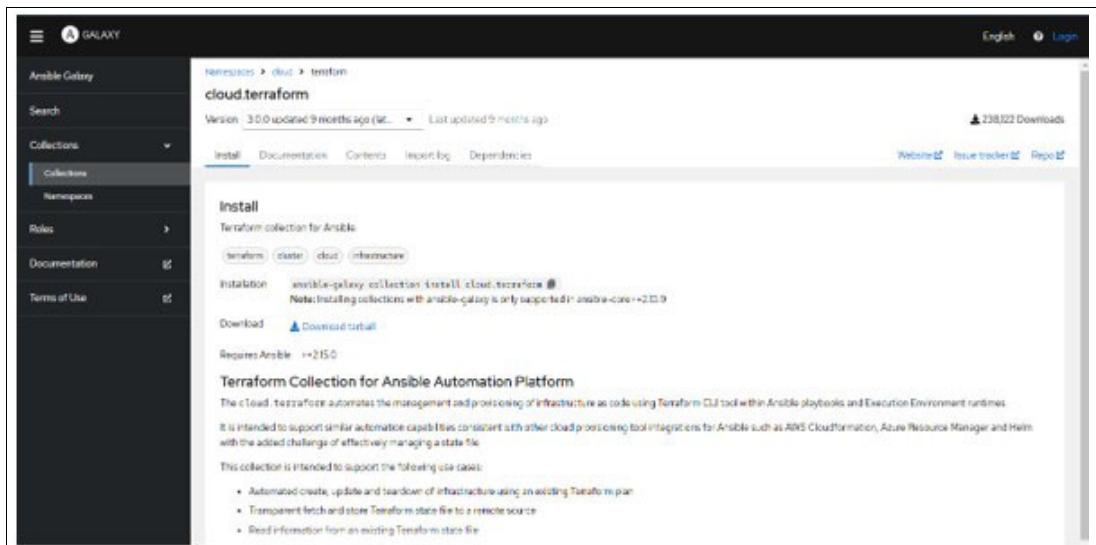


Figure 5-56 Ansible Terraform collection

Integrating Ansible with Terraform enables seamless automation of infrastructure provisioning and configuration management. Terraform defines and creates infrastructure components, such as VMs or cloud resources, and Ansible configures and deploys software on those resources.

5.6 Power Enterprise Pools and Cloud Management Console

IBM Power Enterprise Pools is a technology that enables dynamic sharing of processor and memory activations, and operating system licenses, across a group of IBM Power servers. This solution enhances flexibility, load-balancing, and disaster recovery (DR) planning for IBM Power environments. It provides clients with the capabilities that they need to deploy and manage a private cloud infrastructure effectively.

5.6.1 Power Enterprise Pools 1.0

Power Enterprise Pools 1.0 is a collection of mobile processor and memory activations that can be manually reallocated among servers within the pool. The pool is monitored and managed through the HMC.

5.6.2 Power Enterprise Pools 2.0

IBM Power Private Cloud with Shared Utility Capacity (also known as Power Enterprise Pools 2.0 (PEP 2.0)) provides cloud-like flexibility and efficiency while maintaining the security and control of on-premises infrastructure. This solution enables organizations to share resources across multiple IBM Power servers, optimizing usage through a pay-per-use model with minute-level metering. Processor cores and memory are dynamically allocated as needed, eliminating upfront costs and improving operational agility.

PEP 2.0 allows pooling of CPU and memory resources across a defined set of IBM Power servers, offering greater flexibility in resource usage. Each server in the pool is ordered with a specific number of base processor and memory activations. When a PEP 2.0 pool is configured, these base activations, along with their associated operating system license entitlements, are combined into a shared resource pool. Once the pool is active, resources are instantly available across all participating systems, eliminating the need to manually move mobile resources between servers. Capacity is accessed automatically as demand increases.

If resource usage exceeds the aggregated base capacity, the excess is measured in real time and billed as metered capacity. This usage is either deducted from prepaid capacity credits or invoiced monthly. Resource consumption within PEP 2.0 is tracked and monitored by the IBM Cloud Management Console (CMC).

5.6.3 Cloud Management Console

The IBM CMC for Power provides a consolidated view of the Power servers cloud landscape, regardless of the number of systems or data centers that are involved. It includes an inventory of systems and virtual components, consolidated performance data to optimize usage and performance across all data centers, and aggregated logging information to provide more insights.

CMC runs as a service that is hosted in the IBM Cloud and can be accessed securely at any time. This setup enables system administrators to generate reports and gain insights into their Power cloud deployments. As private and hybrid cloud environments grow, enterprises require deeper visibility into these infrastructures. Tools that offer consolidated information and analytics are essential for helping ensure smooth infrastructure operations.

Figure 5-57 on page 173 illustrates how CMC monitors servers across different pools.

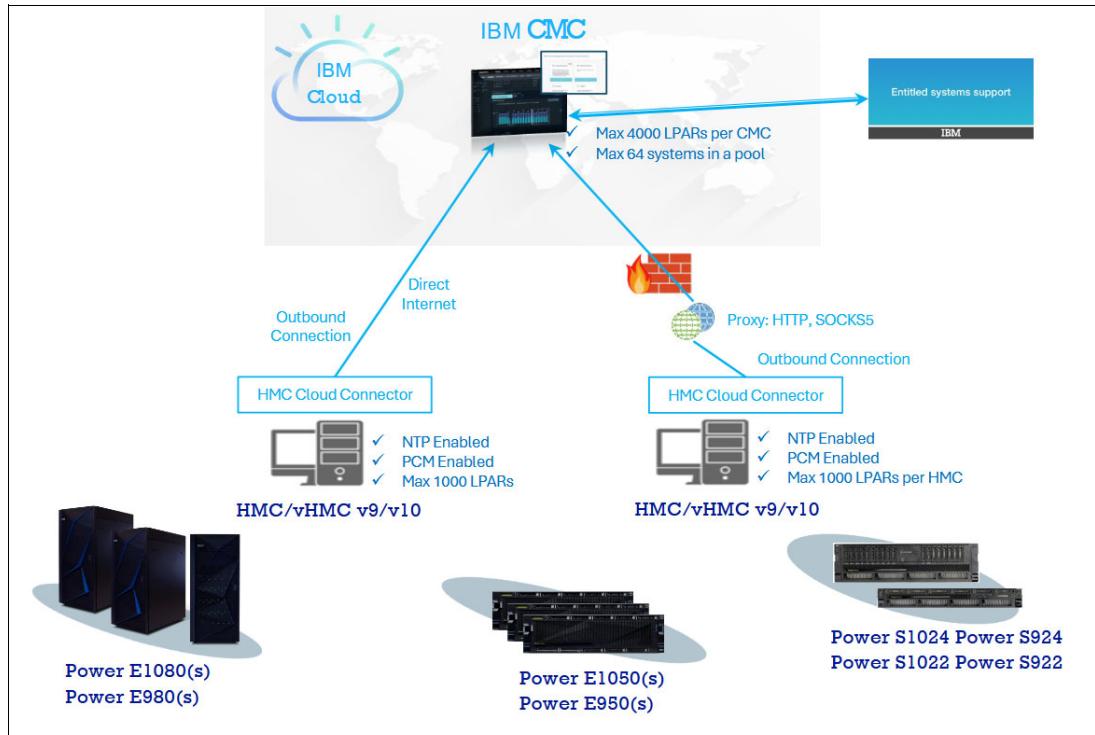


Figure 5-57 IBM Cloud Management Console overview

For more information about Power Enterprise Pools and the CMC, see *IBM Power Systems Private Cloud with Shared Utility Capacity: Featuring Power Enterprise Pools 2.0*, SG24-8478.

5.7 PowerVM networking concepts

PowerVM includes a comprehensive set of networking tools and technologies that enhance flexibility, security, and hardware resource usage. Some of these terms and concepts are unique to the IBM Power Architecture.

Network connectivity in the PowerVM virtual environment is highly flexible. PowerVM virtual networking includes the following technologies:

- ▶ **VNet:** Enables inter-partition communication without requiring a physical network adapter for each partition. If the VNet is bridged, partitions can also communicate with external networks. A VNet is defined by its name or VLAN ID and the associated virtual switch. Use the HMC wizard to configure a VNet.
- ▶ **Virtual Ethernet Adapter:** Allows a client partition to send and receive network traffic without a physical Ethernet adapter.
- ▶ **Virtual switch:** An in-memory, hypervisor-based implementation of a Layer 2 switch. The Power Hypervisor provides a virtual Ethernet switch with IEEE 802.1Q capabilities for VLAN tagging on an IEEE 802.3 Ethernet network. This standard defines VLAN tagging procedures for Ethernet frames and how bridges and switches handle them.

- ▶ VNet bridge: A software adapter that bridges physical and VNets to enable communication. A VNet bridge can be configured for failover or load sharing. The HMC GUI and REST API do not support configuring a VNet bridge with more than two trunk adapters. However, the HMC CLI and VIOS commands can be used to create more trunk adapters with the same VLAN configuration and to assign different priorities across VIOS instances. Once trunk adapters are created, further operations on the VNet bridge must be performed by using the HMC CLI or VIOS commands.
- ▶ Link aggregation device: Also known as an Etherchannel device, this technology aggregates multiple physical Ethernet adapter ports into a single logical device. Link aggregation can follow the Link Aggregation Control Protocol (LACP) that is defined in IEEE 802.1AX or the earlier IEEE 802.3ad standard. It increases bandwidth and resilience by providing higher throughput over a single IP address.

Note: For an overview of PowerVM networking concepts, see [PowerVM networking concepts](#).

5.7.1 Shared Ethernet Adapter technology

PowerVM introduced the SEA to enable the sharing of physical Ethernet adapters across multiple LPARs. This approach provides the flexibility that is required by most clients. However, the SEA-based virtual networking solution introduces a layered software processing impact and multiple data copies from the time that a packet is committed for transmission on the VEA to when it is queued on the physical Network Interface Controller (NIC). The same situation applies to received packets. This impact can lead to performance issues when large volumes of data are transmitted through the SEA.

Performance considerations

Here are some performance and administrative considerations when using the SEA:

- ▶ When creating tagged networks with a one-to-one mapping between VLAN IDs and virtual Ethernet adapters, the range of port VLAN IDs that is used can be large. These VLAN IDs are considered disposable. Coordinate with the network team to define a VLAN ID range that will not be used in the future or that can be reserved specifically for PowerVM networking.
- ▶ Increase the network buffer size for virtual Ethernet adapters, and isolate high-performance workload VLAN IDs on dedicated virtual Ethernet adapters. This approach maximizes buffer allocation for those workloads.
- ▶ The SEA functions as a network bridge. To achieve the required speed and throughput for modern 10 Gbps and 25 Gbps networks, tuning is required on both the VIOS and the client VMs.

Figure 5-58 on page 175 illustrates network adapter sharing across various LPARs.

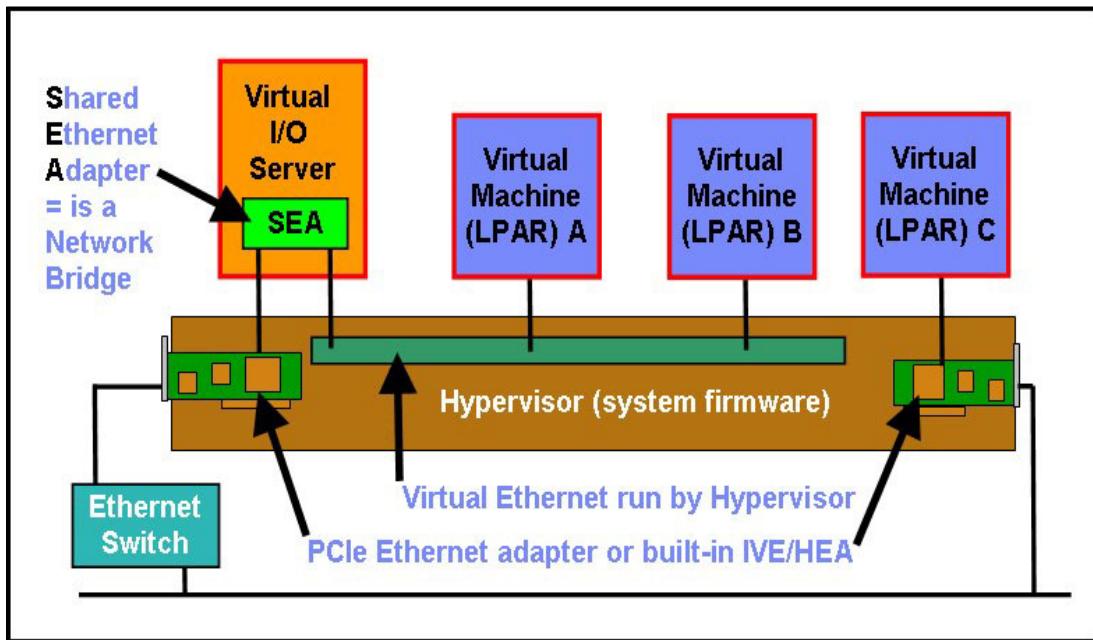


Figure 5-58 SEA as a network bridge⁴

5.7.2 Virtual network interface technology

SR-IOV is an extension to the Peripheral Component Interconnect Express (PCIe) specification that allows multiple LPARs running simultaneously within a single system to share a PCIe device. With the introduction of SR-IOV-capable adapters, the industry gained a method for sharing a network adapter in a PCIe slot across multiple partitions. Before this technology, a PCIe slot could be dedicated only to a single partition and shared by using a VIOS.

An SR-IOV-capable adapter can operate in either dedicated mode (assigned directly to a partition) or shared mode, where the adapter is owned by the hypervisor and used by multiple LPARs simultaneously. In shared mode, a single adapter with multiple physical ports can deliver high-performance networking or data processing capabilities to several partitions through logical ports (LPs), enabling efficient resource sharing while maintaining performance.

SR-IOV technology

An SR-IOV adapter enables the creation of multiple virtual instances of a PCIe function, which is known as Virtual Functions (VFs). Each VF can be independently assigned to an LPAR. SR-IOV VFs operate with minimal software intervention, delivering high performance with low CPU processing impact. A VF is a PCIe function that is defined by the Single Root I/O Virtualization and Sharing specification. It serves as the connection point for an SR-IOV LP.

SR-IOV logical port

An SR-IOV LP is an I/O device that is created for a partition or partition profile by using the HMC. It enables the partition to access a VF on an SR-IOV adapter.

⁴ Source:

<https://www.ibm.com/support/pages/powervm-virtual-ethernet-speed-often-confused-vios-sea-speed>

Virtual Network Interface Controller

A Virtual Network Interface Controller (vNIC) is a PowerVM virtual networking technology that simplifies network management and delivers enterprise-grade capabilities when used with SR-IOV adapters. This high-performance solution provides bandwidth control and QoS at the vNIC level. vNIC technology reduces virtualization processing impact, resulting in lower latency and reduced consumption of server resources such as CPU and memory.

With the introduction of SR-IOV-capable adapters, it is now possible to share a single Ethernet adapter more efficiently, mitigating some of the performance limitations that previously were associated with SEA configurations.

5.7.3 SR-IOV and SEA comparison

Figure 5-59 compares the data flows of SEA and vNIC implementations.

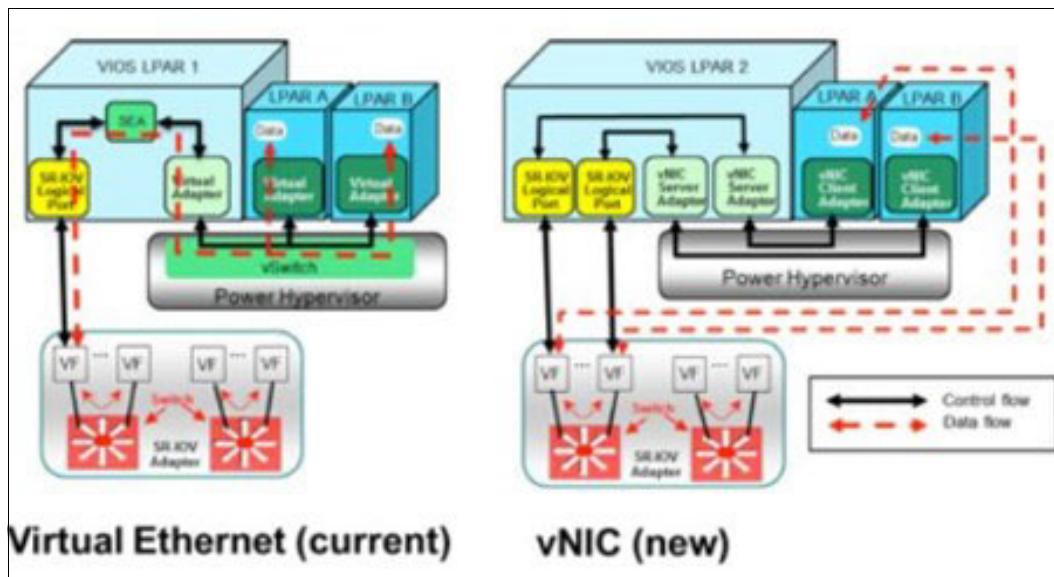


Figure 5-59 SEA data flow compared to vNIC data flow

In the vNIC implementation, control packets flow through the Power Hypervisor, and data packets are delivered directly to the partition. This direct path to the client LPAR reduces software and device latency.

The vNIC is a type of virtual Ethernet adapter that is configured on the LPAR.

Each vNIC is backed by an SR-IOV LP that is available on the VIOS. This configuration helps ensure that the client LPAR remains eligible for LPM. A technology that is called Logical Redirected Direct Memory Access (LRDMA) enables the vNIC to transmit and receive buffers directly to and from the remote SR-IOV LP.

Figure 5-60 on page 177 illustrates the vNIC control and data flow.

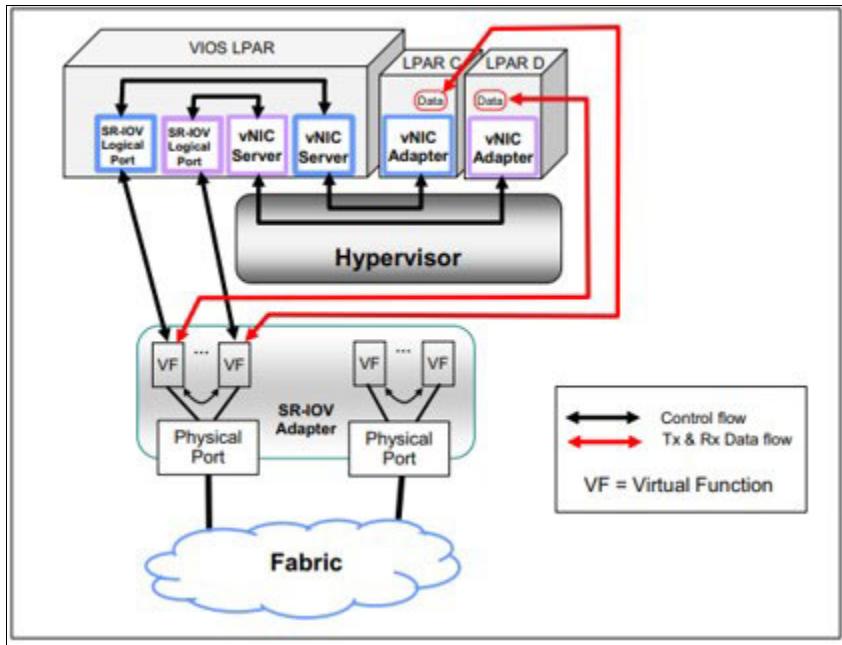


Figure 5-60 Data flow by using a vNIC on an SR-IOV adapter

The key element of the vNIC model is the one-to-one mapping between a vNIC virtual adapter in the client LPAR and the backing SR-IOV LP in the VIOS.

With this model, packet data for transmission (and similarly for reception) is moved directly from the client LPAR memory to the SR-IOV adapter, bypassing the VIOS memory. This approach reduces memory copy processing, resulting in lower latency, and decreases CPU and VIOS memory usage, improving overall efficiency.

Table 5-1 compares the available technologies for network connectivity.

Table 5-1 Comparing network technologies

Technology	LPM	QoS	Direct-access performance	Redundancy options	Server-side failover	Requires VIOS
SR-IOV	No ^a	Yes	Yes	Yes ^b	No	No
vNIC	Yes	Yes	No ^c	Yes ^b	vNIC Failover	Yes
SEA/vEth	Yes	No	No	Yes	SEA Failover	Yes
Hybrid Network Virtualization (HNV)	Yes	Yes	Yes	Yes	No	No

a. SR-IOV can optionally be combined with VIOS and virtual Ethernet to support higher-level virtualization functions such as LPM. However, in this configuration, the client partition does not receive the performance or QoS benefits that typically are associated with SR-IOV.

b. Some limitations apply. For more information, see the [FAQ on link aggregation](#).

c. This approach generally delivers better performance and requires fewer system resources compared to SEA or virtual Ethernet.

SR-IOV adapters

Not all Ethernet adapters are SR-IOV capable. Table 5-2 lists the SR-IOV-capable adapters that are supported on Power10 servers. For information about supported adapters on earlier-generation servers, see this [IBM Community document](#)

Table 5-2 vNIC capable adapters on Power10 servers

SR-IOV-capable network I/O adapters	Feature Codes	Server and attached I/O Expansion Drawer adapters				
		IBM Power S1012	IBM Power S1022, IBM Power S1022s, and IBM Power L1022 (Feature Codes EMX0 and ENZ0)	IBM Power S1014, IBM Power S1024, and IBM Power L1024 (Feature Codes EMX0 and ENZ0)	IBM Power E1050 (Feature Codes EMX0 and ENZ0)	IBM Power E1080 (Feature Codes EMX0 and ENZ0)
PCIe3 2-Port 10GbE NIC and RoCE SR/Cu Adapter ^a	EC2R and EC2S		EC2R (EC2S)	EC2S (EC2S)	EC2S (EC2S)	EC2R (EC2S)
PCIe3 2-Port 25/10GbE NIC and RoCE SR/Cu Adapter	EC2T and EC2U		EC2T (EC2U)	EC2U (EC2U)	EC2U (EC2U)	EC2T (EC2U)
PCIe4 2-port 100/40GbE NIC and RoCE QSFP28 Adapter x16	EC67 and EC66		EC67	EC66	EC66	EC67
PCIe4 x16 2-port 100/40GbE NIC and RoCE QSFP28 Adapter ^b	EC75 and EC76		EC75	EC76	EC76	EC75
PCIe4 2-Port 25/10/1Gb RoCE SFP28 Adapter ^{c,d}	EC71 and EC72	EC71	EC71 (EC72)	EC72 (EC72)	EC72 (EC72)	EC71 (EC72)
PCIe4 4-Port 25/10/1GbE RoCE SFP28 Adapter ^e	EN24 and EN26		(EN26) ^f	EN26 (EN26) ^f	EN26 (EN26) ^f	EN24 (EN26) ^f

a. Withdrawn.

b. SR-IOV support is available on Power10 servers with FW1030.

c. SR-IOV support is available on Power10 servers with FW1050.

d. Supported in the ENZ0 I/O Expansion Drawer and not in the EMX0 I/O Expansion Drawer.

e. SR-IOV support is available on Power10 servers with FW1060.10.

f. Not supported in the EMX0.

Each SR-IOV adapter supports a different number of LPs. Table 5-3 lists the number of LPs or VFs that are supported per adapter on Power servers, along with their connectivity speed options.

Table 5-3 . Number of LPs or VFs that are supported per adapter on Power servers

SR-IOV-capable network I/O adapters	Feature Codes	Physical port link speed	# of LPs per physical port	# of LPs per physical port
PCIe2 4-port (2x10GbE+2x1GbE) SR Optical fiber and RJ45	EN0J, EN0H, EL38, and EL56	1 Gb	4	48
		10 Gb	20	
PCIe2 4-port (2x10GbE+2x1GbE) copper twinax and RJ45	EN0L, EN0K, EL3C, and EL57	1 Gb	4	48
		10 Gb	20	
PCIe2 4-port (2x10GbE+2x1GbE) LR Optical fiber and RJ45	EN0N and EN0M	1 Gb	4	48
		10 Gb	20	
PCIe3 4-port 10GbE SR optical fiber	EN16 and EN15	10 Gb	16	64
PCIe3 4-port 10GbE copper twinax	EN18 and EN17	10 Gb	16	64
PCIe3 LP 2-Port 10 Gb NIC and ROCE SR/Cu Adapter	EC2R and EC2S	10 Gb	40	80
PCIe3 LP 2-Port 25/10Gb NIC&ROCE SR/Cu Adapter	EC2T and EC2U	25/10 Gb	40	80
PCIe3 LP 2-port 100GbE NIC and RoCE QSFP28 Adapter x16	EC3L and EC3M	40/100 Gb	60	120
PCIe4 LP 2-port 100GbE NIC and RoCE QSFP28 Adapter x16	EC66 and EC67	40/100Gb	60	120
PCIe4 x16 2-port 100/40GbE NIC and RoCE QSFP28 Adapter	EC75 and EC76	40/100 Gb	80	160
PCIe4 2-Port 25/10/1 Gb RoCE SFP28 Adapter ^a	EC71 and EC72	1/10/25 Gb	40	80
PCIe4 4-Port 25/10/1 GbE RoCE SFP28 Adapter	EN24 and EN26	1/10/25 Gb	20	80

a. If the physical port is configured for a 1 Gb link speed, the number of LPs per physical port should be limited to a maximum of four.

Restriction: The maximum number of SR-IOV shared mode-enabled adapters per system is 32.

5.7.4 vNIC configuration

The vNIC configuration is performed on the HMC in a single step. This approach is available only through the Enhanced GUI.

When a vNIC adapter is added to an LPAR, the HMC automatically creates all the required components: the SR-IOV logical port and vNIC server adapter on the VIOS, and the vNIC client adapter on the LPAR. No manual configuration is needed on the VIOS.

To access vNIC settings, open the LPAR details in your HMC and select **Virtual NICs** (see Figure 5-61).

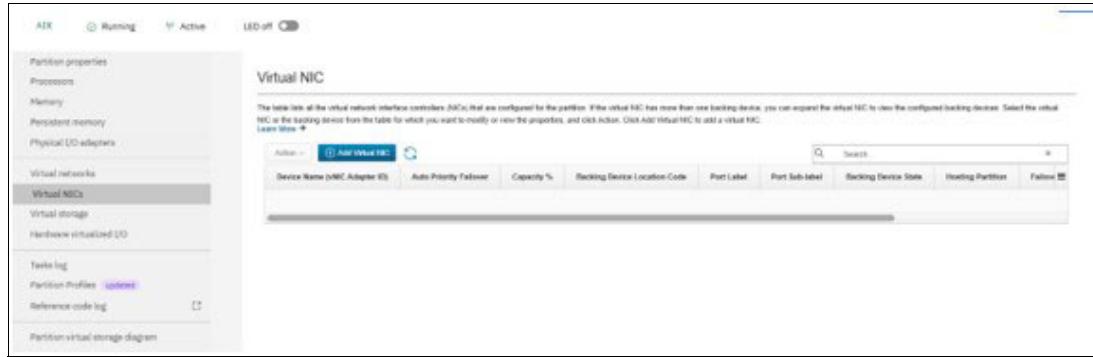


Figure 5-61 HMC Virtual NIC configuration

Add the vNIC to the client LPAR, as shown in Figure 5-62.

When creating a vNIC by using SR-IOV on an IBM Power server, the HMC Enhanced GUI guides you through the required configuration steps.

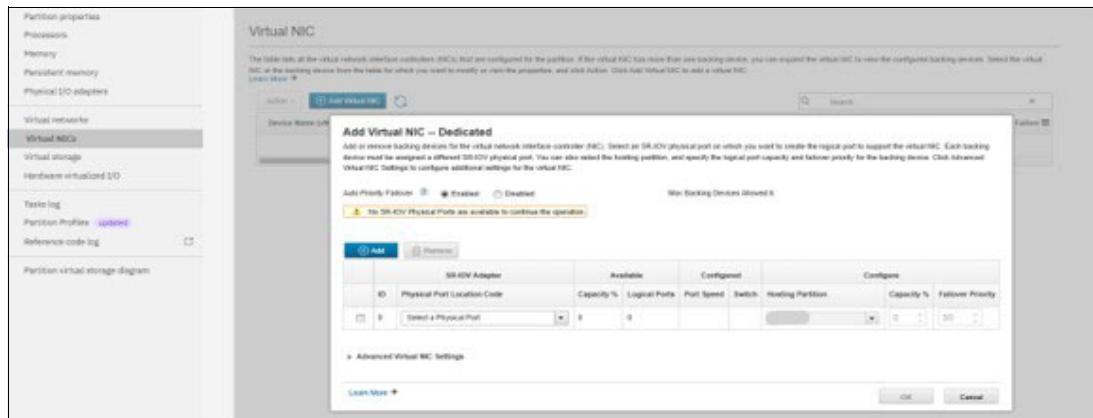


Figure 5-62 Adding a vNIC

The HMC displays a list of VIOS partitions that are eligible to host the SR-IOV port. This selection determines which VIOS manages the vNIC's connection to the physical network. Select a specific port on an available SR-IOV adapter.

The HMC performs a validation check to help ensure that a suitable SR-IOV adapter is available. If no adapter supports SR-IOV, or if the available adapters are not configured correctly, the HMC issues an alert. To enable vNIC functions, the SR-IOV adapter must be set to either *permissive* or *shared* mode. These modes allow the adapter to be virtualized and shared among multiple LPARs, which is essential for SR-IOV operation.

To modify an SR-IOV adapter to shared mode, access the managed system in the HMC, as shown in Figure 5-63 on page 181.

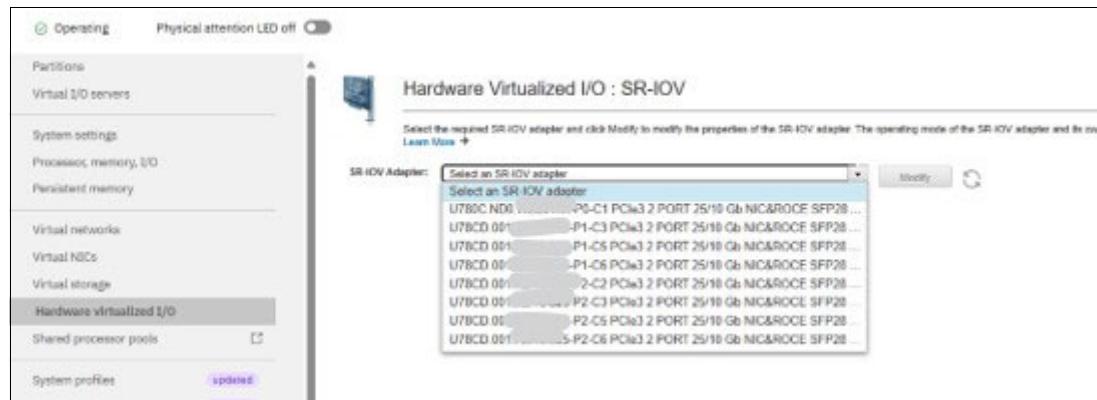


Figure 5-63 Setting an adapter to shared mode

At this stage, you can select a VIOS and an available SR-IOV adapter for the HMC to use when creating the vNIC, as shown in Figure 5-64.

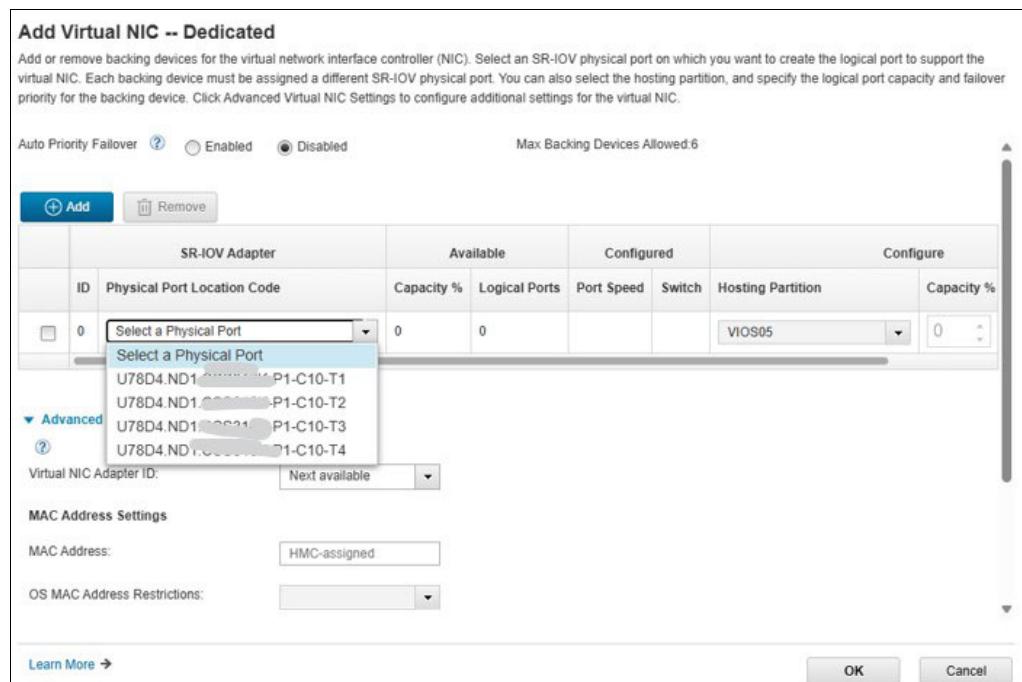


Figure 5-64 Choosing the VIOS and adapter

You can specify a percentage-based capacity allocation to enable QoS management. This feature reserves a defined portion of available bandwidth for the vNIC, helping ensure consistent performance for critical applications.

Also, you can configure a failover priority. This setting is essential in vNIC failover configurations, where multiple vNICs are deployed for redundancy. By assigning priorities, you designate the primary vNIC to handle network traffic under normal conditions while backup vNICs automatically take over in a failure.

Figure 5-65 shows setting these values.

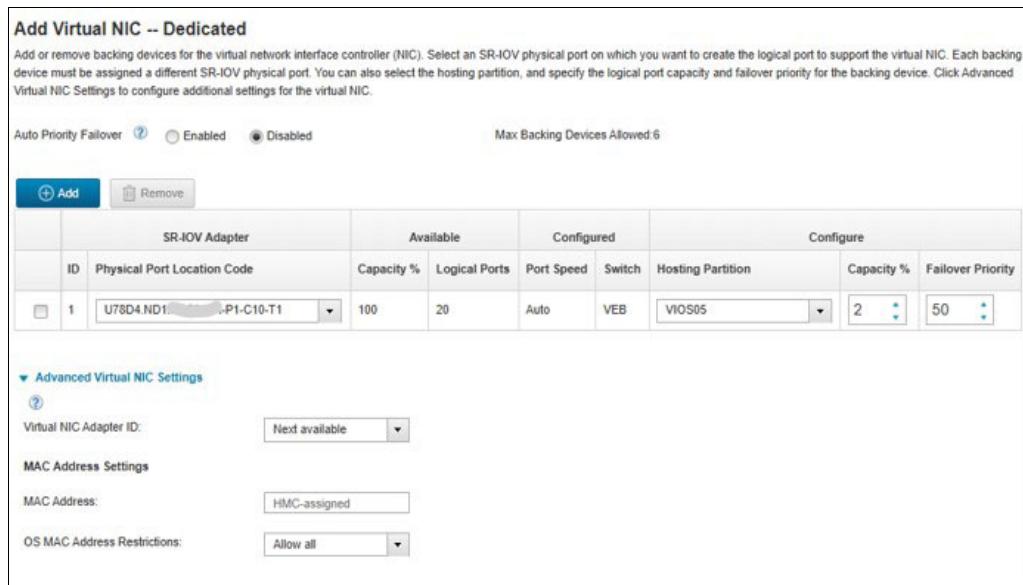


Figure 5-65 Defining details of vNIC

Select the required SR-IOV adapter from the SR-IOV Adapter list to view its details.

For more information about configuring vNICs, see [this technote](#)

Viewing vNIC information

When an adapter is assigned to the VIOS as a vNIC-capable adapter, it is displayed as a Converged Network Adapter VF adapter. However, if the adapter is allocated dynamically, it appears only after a hypervisor update is triggered by shutting down and restarting the VIOS.

Example 5-43 shows the output from a `lsdev` command.

Example 5-43 Output from a `lsdev` command

```
# lsdev -Cc adapter | grep ent
ent0      Available 03-00 PCIe3 10GbE SFP+ SR 4-port Converged Network Adapter (df1020e214100f04)
ent1      Available 03-01 PCIe3 10GbE SFP+ SR 4-port Converged Network Adapter (df1020e214100f04)
ent2      Available 03-02 PCIe3 100/1000 Base-TX 4-port Converged Network Adapter (df1020e214103c04)
ent3      Available 03-03 PCIe3 100/1000 Base-TX 4-port Converged Network Adapter (df1020e214103c04)
ent4      Available 05-00 PCIe3 10GbE SFP+ SR 4-port Converged Network Adapter (df1020e214100f04)
ent5      Available 05-01 PCIe3 10GbE SFP+ SR 4-port Converged Network Adapter (df1020e214100f04)
ent6      Available 05-02 PCIe3 100/1000 Base-TX 4-port Converged Network Adapter (df1020e214103c04)
ent7      Available 05-03 PCIe3 100/1000 Base-TX 4-port Converged Network Adapter (df1020e214103c04)
ent8      Available   Virtual I/O Ethernet Adapter (1-1an)
ent9      Available   Virtual I/O Ethernet Adapter (1-1an)
ent10     Available   Virtual I/O Ethernet Adapter (1-1an)
ent11     Available   Virtual I/O Ethernet Adapter (1-1an)
ent12     Available   Virtual I/O Ethernet Adapter (1-1an)
ent13     Available   Virtual I/O Ethernet Adapter (1-1an)
ent14     Available   Virtual I/O Ethernet Adapter (1-1an)
ent15     Available   Virtual I/O Ethernet Adapter (1-1an)
ent16     Available   Virtual I/O Ethernet Adapter (1-1an)
ent17     Available 08-00 PCIe3 10GbE SFP+ SR 4-port Converged Network Adapter VF (df1028e214100f04)
ent18     Available   Etherchannel / IEEE 802.3ad Link Aggregation
ent19     Available   Shared Ethernet Adapter
ent20     Available   Shared Ethernet Adapter
```

ent21	Available	Virtual I/O Ethernet Adapter (1-1an)
ent22	Available	Virtual I/O Ethernet Adapter (1-1an)
ent23	Available	Virtual I/O Ethernet Adapter (1-1an)

When a VIOS is already using a SEA for network virtualization, the behavior of SR-IOV-capable adapters that are assigned for vNICs differs from traditional adapter visibility.

Specifically, SR-IOV adapters that are dedicated to vNICs on LPARs do not appear in the VIOS profile under the standard I/O adapters list when an SEA is present. Instead, they are visible only in the **SR-IOV Logical Ports** menu. This menu does not require manual configuration because the HMC automates vNIC implementation directly from the client LPAR.

This scenario assumes the use of pure SR-IOV vNICs, without HNV or dedicated LPs from the SR-IOV adapter.

To verify vNIC assignments, use the `lsmmap -vnic` command on the VIOS, as shown in Figure 5-66.



```

$ lsmmap -all -vnic
Name          Physloc
vnicserver0   U9040.MR9.██████████ V1-C32897
                                         32 N/A      N/A
Backing device:ent17
Status:Available
Physloc:U78D4.ND1.██████████-P1-C10-T1-S1
Client device name:N/A
Client device physloc:U9040.MR9.██████████-V32-C5

```

Figure 5-66 Using the `lsmmap` command on the VIOS

This command provides immediate visibility into vNIC details following the dynamic assignment of the SR-IOV-capable adapter to both the VIOS and the LPAR. Crucially, this information is available without requiring a system restart, offering real-time insight into the network configuration.

For more information, use the `lshwres` command through the HMC CLI, as shown in Example 5-44.

Example 5-44 Information from the HMC command line

```

lshwres -m $ms -r virtualio --rsubtype vnic
lpar_name=workstation3-ab76ae0c-0000005c,lpar_id=32,slot_num=5,desired_mode=ded,cu
rr_mode=ded,auto_priority_failover=1,port_vlan_id=10,pvid_priority=0,allowed_vlan_
ids=all,mac_addr=024ff663f705,allowed_os_mac_addrs=all,"backing_devices=sriov/VIOS
06/2/4/0/27010002/2.0/2.0/50/100.0/100.0,sriov/VIOS05/1/1/0/27004002/2.0/2.0/50/
0.0/100.0","backing_device_states=sriov/27010002/0/Link Down,sriov/27004002/1/Link
Down"

```

You can also choose to implement vNICs from the HMC CLI and maintain a record of the configuration in a version-controlled file for auditing or documentation purposes.

vNIC and automation

All vNIC management tasks can be performed through the HMC. However, automation is also available by using the Ansible `ibm.power_hmc` collection, which uses the `cmd` module and the HMC CLI to streamline operations.

Figure 5-67 shows the IBM Power HMC collection as published in Ansible Galaxy.

The screenshot shows the Ansible Galaxy interface. In the top navigation bar, 'GALAXY' is selected. Below it, the path 'Namespaces > ibm > power_hmc > Documentation' is shown. The main content area displays the 'ibm.power_hmc' collection, version 1.1.0, last updated 2 months ago. The 'Documentation' tab is selected. On the left sidebar, under 'Collections', 'ibm.power_hmc' is listed. Under 'Modules (12)', 'hmc_command' is selected and highlighted in blue. The right panel provides detailed documentation for the 'hmc_command' module, including its synopsis, parameters, and examples. The synopsis states: 'Generic module that can execute any HMC CLI command. The given command will be executed on all selected HMC. Information about the HMC CLI commands can be found in the <https://www.ibm.com/docs/en/powerl0/7063-CR1?topic=hmc-commands>' link. The parameters table lists the following:

Parameter	Choices / Defaults	Comments
cmd	string / required	The command to be executed on HMC.
hmc_auth	dictionary / required	Username and Password credential of the HMC.
password	string	Password of the HMC.
username	string / required	Username of the HMC to login.
hmc_host	string / required	The IP address or hostname of the HMC.

Figure 5-67 Ansible HMC collection

5.7.5 vNIC and LPM

vNIC is not the same as HNV and does not use a dedicated SR-IOV port that is assigned directly to a client LPAR. Instead, the vNIC implementation virtualizes the LP by using VFs.

When an LPAR uses a client vNIC adapter, any target system that is involved in LPM must have an SR-IOV adapter in shared mode, with an available LP and sufficient capacity on a physical port.

Note: HNV is built on existing technologies such as AIX Network Interface Backup (NIB) and IBM i Virtual IP Address (VIPA).

vNIC failover

vNIC failover is a configuration in which a vNIC client is backed by multiple devices for redundancy, similar to SEA Failover.

A vNIC client can be configured with up to six backing devices. One device is active while the others remain in standby. If the Power Hypervisor detects that the active backing device is no longer operational, it initiates a failover to the most preferred operational device, which is determined by the lowest Failover Priority value.

Power10 servers, along with supported versions of HMC, system firmware, and OSs, include support for vNIC failover.

A vNIC backing device consists of an SR-IOV adapter physical port in shared mode, an SR-IOV LP, a vNIC server virtual adapter, and a VIOS.

During configuration, the user selects a VIOS and a physical port on an SR-IOV adapter. When the backing device is instantiated, the HMC creates the vNIC server adapter and LP on the selected VIOS. Once created, the backing device is available to the vNIC client as either an active or standby device. Each LP and vNIC server adapter is associated with a single vNIC client.

You can modify the configuration of the Virtual NIC attached to the LPAR in the HMC, as shown in Figure 5-68.

Action	Priority Failover	Capacity %	Backing Device Location Code	Port Label	Port Sub-label	Backing Device Status
	2	U78D4.NL	C10-T1-S1 (ent17)			Link Down (Active)

Figure 5-68 Virtual NIC configuration

When viewing the vNIC backing devices, you can see the vNICs that are assigned to the LPAR, as shown in Figure 5-69.

Backing Device Location Code	Port Label	Port Sub-label	Capacity %	Backing Device Status
U78D4.NL			2.0	Link Down (Active)

Figure 5-69 Viewing the available backing devices

You can add a backing device to the existing vNIC by selecting **Modify backing device**, as shown in Figure 5-70.

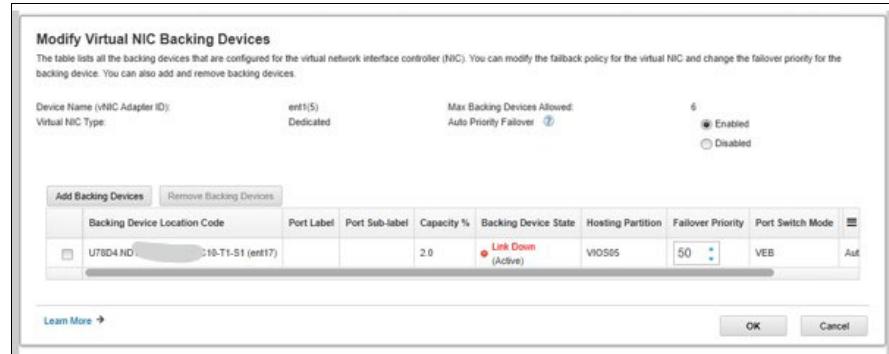


Figure 5-70 Adding a backing device to a vNIC

When multiple SR-IOV-capable adapters are configured in shared mode on the managed system, you can select an alternative VIOS as the hosting partition for an extra vNIC backing device, as shown in Figure 5-71.



Figure 5-71 Selecting a VIOS for failover

The Power Hypervisor manages failover by using the Failover Priority attribute of the vNIC. When this attribute is set to **Automatic**, the hypervisor determines which VIOS has priority for handling network traffic and selects the appropriate VIOS for failover to maintain availability, as shown in Figure 5-72.

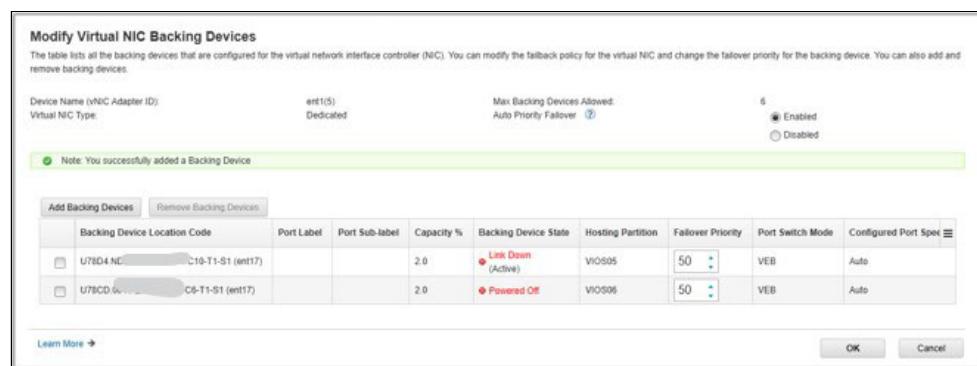


Figure 5-72 Setting the failover priority

You can modify the priority of each vNIC dynamically after configuration by selecting **Modify Virtual NIC Backing Devices** in the HMC, as shown in Figure 5-73.

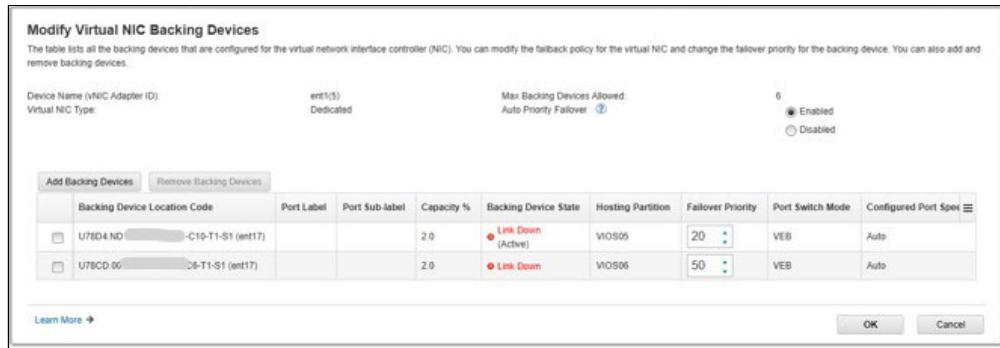


Figure 5-73 Modifying the vNIC priority

Information about vNIC backing devices is also available by using the **entstat** command. The output includes a “Server Information” section, which displays only the VIOS that is providing the SR-IOV path. This behavior is illustrated in Example 5-45.

Example 5-45 Displaying the vNIC information with entstat

```
/>entstat -d ent1 | grep -p "Server Information"
Server Information:
    LPAR ID: 1
    LPAR Name: VIOS05
    VNIC Server: vnicserver0
    Backing Device: ent17
    Backing Device Location: U78D4.ND1.XXXXXX-P1-C10-T1-S2
```

The vNIC is presented as a vnic device.

```
/>lsdev -Cc adapter
ent0 Available      Virtual I/O Ethernet Adapter (l-lan)
ent1 Available      Virtual NIC Client Adapter (vnic)
fcs0 Available C3-T1 Virtual Fibre Channel Client Adapter
fcs1 Available C4-T1 Virtual Fibre Channel Client Adapter
hdrypt Available   Data encryption
pkcs11 Available ACF/PKCS#11 Device
vs0 Available       LPAR Virtual Serial Adapter
vscsi0 Available  Virtual SCSI Client Adapter
root@workstation3:/>
```

vNIC VIOS resource requirements

Table 5-4 provides the estimated memory usage by the Power Hypervisor and VIOS for both vNIC and vNIC failover configurations. Actual memory usage might vary depending on the system configuration and workload.

Table 5-4 Estimated resource allocation for vNIC

Adapter Feature Code	Hypervisor memory per adapter	Hypervisor memory per vNIC client	Hypervisor memory per vNIC backing device	VIOS memory per vNIC backing device
EN0H, EN0J, EN0K, EN0L, EN0M, EN0N, EN15, EN16, EN17, and EN18	160 MB	9 MB	0.7 MB	7.5 MB
EC3L and EC3M	3.7 GB	9 MB	0.7 MB	25 MB
EC2T, EC2U, EC2R, and EC2S	2.9 GB	9 MB	0.7 MB	25 MB
EC66 and EC67	5.4 GB	9 MB	0.7 MB	25 MB
EC75 and EC76	5.9 GB	9 MB	0.7 MB	25 MB
EC71 and EC72	2.9 GB	9 MB	0.7 MB	25 MB
EN24 and EN26	4.5 GB	9 MB	0.7 MB	25 MB

CPU usage on a VIOS that is handling vNIC traffic is highly variable and depends on the nature of the network workload. As a general guideline, approximately 0.7 additional VIOS processor cores should be allocated per 10 Gbps of bandwidth when supporting peak throughput scenarios involving large packets. However, this estimate serves only as a baseline.

Workloads that are characterized by high message rates and small packet sizes require more CPU resources due to the increased processing impact that is associated with handling a larger volume of individual packets. Therefore, careful monitoring and capacity planning are essential to help ensure that the VIOS can manage network traffic efficiently and avoid becoming a performance bottleneck.



How to modernize your applications

Modernizing applications on IBM Power involves transforming existing assets to succeed in a rapidly evolving technological landscape. As you modernize your applications running on IBM Power servers, focus on the following areas:

- ▶ Use the strengths of your existing applications. IBM Power servers are known for their reliability, performance, and security. Modernization efforts should preserve these strengths while introducing new capabilities.
- ▶ Embrace hybrid cloud capabilities. Modernization often includes integrating on-premises Power servers with cloud-based resources to create a hybrid cloud environment.
- ▶ Use containerization and Kubernetes. Technologies such as Red Hat OpenShift support containerization, enabling applications to be divided into smaller, more manageable components. This approach enhances agility and scalability.
- ▶ Adopt cloud-native development for new functions. Modernization encourages cloud-native practices, including microservices, application programming interfaces (APIs), and DevOps methodologies.
- ▶ Integrate artificial intelligence (AI). IBM Power is suitable for AI workloads. Focus on incorporating AI capabilities into existing applications.
- ▶ Prioritize the data. IBM Power provides excellent performance and reliability for data access. Modern applications must handle and analyze large volumes of data.

The following topics are described in this chapter:

- ▶ 6.1, “Modernizing systems through domain segmentation” on page 190
- ▶ 6.2, “Programming languages” on page 193
- ▶ 6.3, “Database technologies” on page 198
- ▶ 6.4, “CI/CD tools” on page 201
- ▶ 6.5, “Support for .NET on IBM Power” on page 202
- ▶ 6.6, “Kubernetes” on page 203

6.1 Modernizing systems through domain segmentation

This section explores the challenges of modernizing systems and presents domain segmentation as an effective strategy for overcoming these obstacles. It describes the principles of domain-driven design (DDD), its role in decomposing monolithic systems, and the benefits of this approach for improving maintainability, scalability, and adaptability.

Modernizing systems is a critical challenge for many organizations. These systems, often built with monolithic architectures, struggle to meet the rapidly evolving demands of today's digital environment. They are typically characterized by tight coupling, limited scalability, slow development cycles, and difficulty adapting to change.

Domain segmentation, inspired by DDD principles, offers a practical solution to these challenges. It involves dividing the system into smaller, self-contained units, each with distinct business logic and data. This approach establishes clear boundaries between contexts and implements each bounded context as a separate microservice, enabling independent development, deployment, and scaling.

Domain segmentation enhances maintainability by breaking the system into manageable units. Changes can be made to individual domains with minimal impact on other parts of the system. Microservices can be independently scaled to meet the specific demands of each domain, optimizing resource usage. The ability to deploy and update microservices independently enables faster delivery of new features and quicker responses to changing business needs. By isolating failures within individual domains, domain segmentation also improves system resilience and reduces the impact of outages.

Implementing domain segmentation involves the following tasks:

- ▶ Identifying bounded contexts by analyzing the system to find natural boundaries and areas of high cohesion and low coupling
- ▶ Defining clear communication protocols and data exchange formats between microservices
- ▶ Gradually developing and deploying microservices, starting with the most critical or high-impact areas
- ▶ Continuously monitoring and refining domain boundaries as needed

Key challenges include maintaining data consistency across microservices, managing the complexity of a distributed system, and helping ensure thorough testing and debugging of interactions between services.

Despite these challenges, domain segmentation offers a promising path for modernizing systems. By decomposing monolithic architectures into smaller, more agile units, organizations can improve scalability, resilience, and adaptability while reducing development and maintenance costs.

Domain Model Definition: A blueprint for microservices

The Domain Model Definition (DMD) is a pragmatic approach to structuring complex software systems, particularly ones that adopt a microservices architecture. It serves as a blueprint for decomposing a system into smaller, more manageable domains.

Traditional data models often suffer from complex interconnections that hinder maintainability and adaptability. Designing and implementing independent microservices requires a clear understanding of domain boundaries and interactions.

The DMD promotes the division of a system into distinct, self-contained domains, each with its own data and logic. It is defined by using flexible formats such as YAML, XML, or JSON, making it independent of specific programming languages, databases, or development tools. Mature design tools typically use UML. The DMD is not a static artifact because it evolves iteratively as the system grows and changes, accommodating new requirements and addressing unforeseen challenges.

By breaking down the system into smaller, focused domains, the DMD improves maintainability and reduces the risk of unintended side effects. Its platform-neutral nature allows for greater flexibility in technology choices and adaptation to evolving business needs. The DMD provides a clear foundation for designing and implementing microservices, helping ensure that they are defined, independent, and aligned with business requirements.

The implementation process includes the following tasks:

- ▶ Gathering relevant information about the system, including data models, existing applications, and business requirements
- ▶ Identifying and defining distinct domains based on business logic, data dependencies, and team responsibilities
- ▶ Converting defined domains into implementation artifacts such as microservices, data models, and APIs
- ▶ Continuously monitoring and refining the DMD to accommodate changes in requirements, technology, or organizational structure

Challenges include maintaining data consistency across microservices, managing the complexity of distributed systems, and helping ensure thorough testing and debugging of service interactions.

The DMD is a valuable tool for organizations adopting a microservices architecture. By providing a structured and flexible approach to domain modeling, it enables development teams to build robust, scalable, and maintainable systems that meet evolving business needs.

DMD: A deeper dive

The DMD is more than a simple data structure. It is a strategic blueprint for building complex software systems, particularly ones that are adopting a microservices architecture.

Rooted in DDD principles, the DMD emphasizes understanding the core business domain, identifying key concepts and their relationships, and converting this knowledge into a software model. It helps define bounded contexts, which are self-contained units within the system that have their own language and rules. This modular approach reduces complexity and promotes maintainability.

The DMD encourages the usage of a shared language between business stakeholders and developers. This common vocabulary helps ensure clear communication and minimizes misunderstandings.

A comprehensive DMD includes the following components:

- ▶ A conceptual representation of the core business domain, capturing key entities, attributes, and relationships.
- ▶ Explicit boundaries that separate different domains, helping isolate changes and reduce the impact of modifications.

- ▶ A representation of data storage and access within each domain, including database schemas, data structures, and APIs.
- ▶ Service specifications that define how services interact with each domain, including interfaces, data formats, and communication protocols.

The DMD fosters better collaboration between business and development teams. By breaking the system into smaller, manageable domains, it improves maintainability and reduces the risk of unintended side effects. Its modular structure enables faster, more flexible development and deployment of new features. The clear separation of concerns also simplifies testing at both the component and system levels.

The DMD can be represented by using various formats, including UML diagrams, textual descriptions, or specialized modeling tools. It is an evolving artifact that should be continuously refined as the system and its requirements change. Using tools and automation can streamline code generation, service deployment, and DMD management.

As an example, in an e-commerce system, the DMD might define separate domains such as:

- ▶ Product Catalog
Manages product information, pricing, and inventory.
- ▶ Order Management
Handles order processing, fulfillment, and shipping.
- ▶ Customer Relationship Management (CRM)
Manages customer accounts, preferences, and interactions.

Each domain includes its own data model, services, and potentially a dedicated development team. This structure enables independent development, deployment, and scaling of each domain, supporting agility and maintainability across the system.

Modernization strategy: Gradual microservices with a shared database

A phased approach to microservices modernization helps mitigate the risk of overwhelming the project and reduces disruption to existing systems.

Begin by developing new microservices that interact with the existing monolithic database. This approach minimizes the initial impact on the system and enables a gradual migration. Frameworks such as Spring Boot are suitable for building microservices and integrating with existing databases.

Enforce strict adherence to domain boundaries. Microservices must not directly access data within another service's domain. All inter-service communication should occur through well-defined APIs to help ensure loose coupling, improve maintainability, and support independent scaling and deployment.

This phased approach offers several advantages:

- ▶ Reduced risk: Minimizes disruption and allows for incremental improvements.
- ▶ Faster time-to-market: Enables quicker delivery of new features and functions.
- ▶ Improved agility: Provides greater flexibility to adapt to changing business requirements.

As modernization progresses, consider strategies for database refactoring, such as data partitioning or creating dedicated databases for specific microservices. These enhancements further improve performance, scalability, and maintainability.

Conclusion

The DMD is a powerful tool for building complex software systems, especially for systems that incorporate a microservices architecture. By providing a structured and well-defined framework for understanding and modeling the business domain, the DMD enables development teams to create systems that are maintainable, adaptable, and aligned with business requirements.

6.2 Programming languages

Selecting the appropriate programming language depends on several factors, including project requirements, performance needs, scalability, developer expertise, and the availability of supporting libraries and tools. Key considerations include the following ones:

- ▶ Project requirements

Different projects demand different capabilities. For example, Python is suitable for data science, machine learning (ML), and rapid prototyping due to its simplicity and extensive libraries such as TensorFlow and Pandas. JavaScript is ideal for web development, with frameworks like React and Angular for front-end development and Node.js for back-end services.

- ▶ Performance and scalability

For applications requiring high performance and scalability, languages such as Go and Rust are strong candidates. Go offers an efficient concurrency model, making it suitable for cloud services and back-end systems. Rust provides memory safety and performance, making it ideal for system-level programming and applications requiring fine-grained control over hardware resources.

- ▶ Developer expertise

The skill set of your development team is a critical factor. If your team is experienced in a particular language, using that expertise can improve efficiency. For example, teams that are familiar with RPG can benefit from using modern RPG on IBM i to maintain and modernize systems.

- ▶ Ecosystem and libraries

The availability of libraries and frameworks can influence language choice. Python supports a wide range of applications, from web development to scientific computing. JavaScript integrates well with numerous web technologies. Go and Rust also offer growing ecosystems that support modern development needs.

- ▶ Integration with existing systems

Consider how well the language integrates with your current infrastructure. For example, modern RPG on IBM i can interface seamlessly with other languages and tools, making it a strong option for modernizing systems while maintaining reliability.

- ▶ Community and support

A strong developer community provides valuable resources and support. Languages such as Python and JavaScript benefit from large, active communities that contribute tutorials, documentation, and third-party libraries.

Ultimately, the correct programming language depends on balancing these factors to meet your specific needs. Often, using a combination of languages enables you to use the strengths of each for different parts of your application.

6.2.1 Integrating traditional and modern technologies

Combining traditional programming languages such as RPG and COBOL with modern development approaches on IBM Power servers is a strategic way to use existing investments while adopting newer technologies. One effective method is to extend applications by using microservices and modern languages. This approach enhances capabilities without requiring a complete rewrite.

However, this hybrid modernization strategy presents ongoing challenges, particularly in maintaining and updating existing RPG and COBOL codebases. Here are some key issues:

- ▶ Shrinking talent pool

As veteran RPG and COBOL developers retire, the number of experienced professionals continues to decline. This expertise gap makes it increasingly difficult to maintain and evolve earlier systems.

- ▶ Attracting new talent

Modern developers often prefer languages such as Python, JavaScript, and Go, which offer contemporary features and broader applicability. This preference makes it more difficult to recruit new talent for IBM i environments.

- ▶ Training and education

Bridging the skills gap requires significant investment in training programs. Organizations must train current staff and educate new hires about traditional languages and IBM i platform specifics. Although time-consuming and costly, this effort is essential for successful modernization.

- ▶ Integration with modern technologies

Modernizing IBM Power servers often involves integrating earlier applications with cloud platforms, AI-driven analytics, and web interfaces. Developers must be proficient in both traditional languages and modern tools to help ensure seamless operation and interoperability.

Tools to aid migration

Tools such as IBM Rational® Developer for i (IBM RDi) and IBM i Modernization Engine for Lifecycle Integration (Merlin) help convert fixed-form RPG code to modern free-form RPG. This conversion makes earlier code more accessible, maintainable, and aligned with contemporary development practices.

The role of generative AI

Emerging AI tools such as IBM RPG Code Assistant are transforming modernization workflows. These tools help developers interpret earlier code, generate new functions from natural language prompts, and automatically create test cases. This automation accelerates development and reduces manual effort.

Conclusion

Addressing these challenges is essential for organizations that aim to modernize IBM Power servers without compromising the performance and reliability of earlier applications. By using the strengths of traditional technologies alongside modern innovations, organizations can adopt a balanced, future-ready approach to IT transformation.

6.2.2 Choosing a modern language for your project

Modern programming languages offer distinct advantages that are tailored to various domains of software development. Understanding their strengths helps organizations choose the right tools for specific project requirements.

- ▶ Python

Python is one of the most widely used modern programming languages, and it is valued for its simplicity, readability, and extensive ecosystem of libraries and frameworks. Its clean syntax enables developers to focus on solving problems rather than managing complex language rules. Python is commonly used in data science, AI, web development, and automation. Libraries such as TensorFlow, Django, and Pandas make Python a strong choice for scientific computing, ML, and web application development. Its versatility and large community support make Python ideal for both rapid prototyping and production-grade solutions.

- ▶ JavaScript

Initially popular for client-side web development, JavaScript is now essential for building full-stack applications, largely due to the rise of Node.js. JavaScript enables the creation of dynamic, responsive user interfaces and serves as the foundation of modern web development. Frameworks such as React, Angular, and Vue.js support front-end development, and Node.js extends JavaScript to the server side for scalable network applications. Its asynchronous, non-blocking nature makes JavaScript suitable for high-performance, real-time applications such as chat platforms and live updates. Integration with a wide range of web technologies further solidifies its role as a core skill for web developers.

- ▶ Go (Golang)

Go is a statically typed, compiled language that is developed by Google, and it is known for its simplicity, performance, and efficiency in handling concurrency. It is widely used in cloud infrastructure, back-end services, and microservices architecture. The Go concurrency model, based on Go routines and channels, enables developers to build scalable and concurrent applications. Its fast compilation times and minimalistic design make it a preferred language for cloud-native applications, APIs, and infrastructure automation projects.

- ▶ Rust

Rust is a systems programming language that is designed for safety, performance, and concurrency. It provides memory safety without relying on garbage collection, making it ideal for performance-critical applications such as game engines, embedded systems, and operating systems. The Rust ownership model prevents common bugs like null pointer dereferencing and data races, which are prevalent in low-level languages such as C and C++. Its compile-time safety helps ensure that Rust is the first choice for developers who require fine-grained control over system resources while maintaining robust and secure software.

- ▶ Modern RPG on IBM i

Modern RPG evolved from its origins as a column-based language into a free-form language that supports contemporary business needs. This transformation helps ensure that IBM i remains a robust platform for application development. Modern RPG supports the IBM Integrated Language Environment® (ILE), enabling seamless integration with other languages and tools. Tools such as IBM RDi and Merlin automate the conversion of fixed-form RPG code to free-form RPG, making earlier code more accessible to new developers. These modernization efforts help maintain the relevance and power of IBM i servers in today's business environment.

Conclusion

Languages such as Python, JavaScript, Go, Rust, and Modern RPG offer unique strengths that are aligned with specific development needs. Python excels in usability and scientific computing, JavaScript dominates web development, Go supports scalable back-end systems, Rust helps ensure safety and performance in system-level programming, and Modern RPG sustains the reliability of IBM i platforms. Selecting the appropriate language depends on project goals such as performance, scalability, and simplicity of development. Together, these languages reflect the evolving demands of the software industry and serve as essential tools for developers across diverse sectors.

6.2.3 Integrated development environments

Using an integrated development environment (IDE) offers several advantages that enhance developer productivity and software quality. IDEs provide a cohesive set of tools that support all stages of the development lifecycle.

Here are some key benefits of IDEs:

- ▶ Code editing and navigation

IDEs offer advanced features such as syntax highlighting, code completion, and real-time error detection. These capabilities help developers write code more efficiently and with fewer errors. Navigation tools enable quick access to different parts of the codebase, improving workflow and reducing context switching.

- ▶ Debugging

Integrated debugging tools enable developers to set breakpoints, inspect variables, and step through code execution. This function simplifies the process of identifying and resolving issues, leading to more reliable software.

- ▶ Version control integration

Many IDEs integrate with version control systems such as Git. This integration allows developers to manage repositories, commit changes, and handle branching and merging directly within the IDE, streamlining collaboration and code management.

- ▶ Build and deployment tools

IDEs often include features for compiling, packaging, and deploying applications. These tools automate repetitive tasks, reduce the risk of errors, and accelerate the delivery process.

- ▶ Project management

IDEs help organize code files, libraries, and resources, making it simpler to manage large projects. This structure supports team collaboration and improves maintainability.

- ▶ Extensibility

Most IDEs support plug-ins and extensions that add functions or integrate with external tools. Developers can tailor their environment to specific needs, such as supporting more languages or enhancing productivity.

- ▶ Consistency

IDEs promote a consistent development environment across teams and projects. This consistency helps enforce coding standards and reduces environment-related issues.

- ▶ Productivity

By consolidating essential tools into a single interface, IDEs minimize the need to switch between applications. This integration helps developers stay focused and work more efficiently.

IDEs play a critical role in developing and modernizing applications on IBM Power servers, including IBM i, AIX, and Linux. They also support development on platforms such as Red Hat OpenShift. Tools like IBM RDi and Merlin provide specialized capabilities for working with traditional and modern languages, enabling seamless integration and modernization of earlier systems.

IBM i

IBM RDi is the primary IDE for IBM i application development. Built on the Eclipse platform, IBM RDi provides a comprehensive set of tools for creating, maintaining, and modernizing applications on IBM i. It supports multiple programming languages, including RPG, COBOL, Java, and C/C++.

IBM RDi offers advanced features such as language-aware source editing, visual analysis tools, and integrated compile error feedback. These capabilities streamline the development process and improve code quality. Also, IBM RDi integrates with IBM Engineering Workflow Management to support application lifecycle management, enabling better collaboration and project tracking.

AIX

For AIX development, commonly used IDEs include IBM Open XL C/C++ and SlickEdit. IBM Open XL C/C++ is a next-generation compiler that is designed for developing and maintaining C/C++ applications on IBM Power platforms. It incorporates LVM and Clang compiler infrastructure to optimize performance and maximize hardware usage.

SlickEdit is another choice among AIX developers. It offers robust editing capabilities and supports a wide range of programming languages, including C/C++, Java, and Python. Its flexibility and powerful editing tools make it for complex development environments.

Linux

Linux supports a wide range of IDEs that are tailored to various development needs. Visual Studio Code (VS Code) is one of the most used choices due to its flexibility, extensive library of extensions, and strong Git integration. It supports multiple programming languages and offers features such as IntelliSense for code completion and integrated debugging.

IntelliJ IDEA is another powerful IDE that is suitable for Java development. It provides intelligent code analysis, smart code completion, and seamless integration with version control systems. Its robust feature set makes it a preferred tool for enterprise-grade application development on Linux platforms.

Red Hat OpenShift

Red Hat OpenShift Dev Spaces is a cloud-based development environment that is built on the Eclipse Che project. It uses Kubernetes and containers to deliver a consistent, secure, and zero-configuration workspace for developers. Dev Spaces supports IDEs such as VS Code and JetBrains IntelliJ IDEA, enabling developers to code, build, test, and run applications directly in the browser.

By defining development environments as code, Dev Spaces eliminates the “works on my machine” problem and helps ensure consistency and reproducibility across teams. Red Hat also provides IDE extensions for Red Hat OpenShift, enhancing the development experience by integrating platform-specific functions directly into supported IDEs.

Conclusion

Selecting the appropriate IDE for IBM Power servers depends on the specific requirements of the development environment and target platform:

- ▶ IBM i: IBM RDi is the preferred IDE, offering specialized tools for RPG and COBOL development.
- ▶ AIX: IBM Open XL C/C++ and SlickEdit provide robust support for C/C++ development on AIX systems.
- ▶ Linux: Developers benefit from versatile IDEs such as VS Code and IntelliJ IDEA.
- ▶ Red Hat OpenShift: Red Hat OpenShift Dev Spaces delivers a modern, cloud-based development environment that integrates seamlessly with Kubernetes and supports browser-based development.

Each IDE offers unique capabilities that are tailored to the needs of its platform, helping developers streamline workflows, maintain code quality, and accelerate modernization efforts.

6.3 Database technologies

IBM Power servers are renowned for their performance, reliability, and scalability, which are qualities that make them ideal platforms for database workloads. Although many organizations have traditionally relied on IBM Db2, the IBM Power platform evolved to support a broad ecosystem of modern, open-source, and NoSQL databases.

6.3.1 Db2: The enterprise-grade relational backbone

IBM Db2 is a powerful relational database management system (RDBMS) that is optimized for IBM Power Architecture, particularly on AIX and IBM i. It excels in online transaction processing (OLTP), complex analytics, and enterprise integration. With features such as high availability (HA), disaster recovery (DR), and mature tools, Db2 remains a preferred choice for enterprise-grade workloads.

Although Db2 adheres to a traditional relational model, it evolved to meet modern application demands. It now supports JSON document storage, RESTful APIs, and integration with big data frameworks such as Apache Spark and Hadoop. These enhancements allow Db2 to function as both a traditional RDBMS and a NoSQL-style engine, enabling hybrid architectures that combine structured and unstructured data models.

Db2 supports both transactional and analytical workloads, offering capabilities such as AI-powered query optimization, continuous availability, and robust security. It can be deployed on-premises, in the cloud, or in hybrid environments, providing the flexibility that is needed to support diverse business requirements.

6.3.2 Modern databases

Modern databases represent an evolution in data management that is driven by a need to handle the volume, velocity, and variety of data that is generated in today's digital landscape. Unlike traditional relational databases, which rely on structured tables and rigid schemas, modern databases offer greater flexibility and scalability. They include a broad range of technologies, such as NoSQL, NewSQL, and cloud-native databases, each of which is designed to address specific data management challenges.

One of the key advantages of modern databases is their ability to scale horizontally. Instead of relying on more powerful hardware (vertical scaling), they can distribute data and workload across multiple servers, enabling them to handle massive datasets and high-traffic loads. Modern databases also offer flexible schema, allowing developers to work with semi-structured and unstructured data, such as JSON documents and time-series data. This flexibility simplifies data modeling and enables faster development cycles.

Modern databases also excel in supporting a wide range of use cases.

- ▶ NoSQL databases (for example, MongoDB and Cassandra) are ideal for applications that require high throughput and low latency, such as real-time analytics, Internet of Things (IoT) platforms, and mobile back ends.
- ▶ NewSQL databases (for example, CockroachDB and YugabyteDB) combine the scalability of NoSQL with the ACID compliance of traditional relational databases, making them suitable for financial systems and e-commerce platforms.
- ▶ Cloud-native databases (for example, Amazon DynamoDB, Azure Cosmos DB, and Google Cloud Spanner) offer fully managed, globally distributed services that simplify administration and help ensure HA.

Modern databases provide a powerful toolkit for managing data in complex and dynamic environments. Their scalability, flexibility, and broad capabilities enable organizations to build innovative applications, extract insights, and drive business growth. By understanding the strengths of each database type, businesses can make informed decisions and select the right tools to unlock the full potential of their data.

6.3.3 Conclusion and comparison

IBM Power servers are no longer limited to traditional relational databases because they evolved into open platforms that support a wide range of database technologies. From OLTP systems to real-time NoSQL stores and containerized, cloud-native databases, IBM Power provides the performance, scalability, and flexibility that are needed for modern data workloads.

Organizations can run transactional systems by using Db2 for i; support real-time web services with MongoDB; or perform large-scale analytics with PostgreSQL or Cassandra. IBM Power supports diverse deployment models, including traditional on-premises environments on AIX, IBM i, or Linux; virtualized environments that use PowerVM or IBM PowerVS; and containerized deployments with Red Hat OpenShift. These options enable organizations to modernize their data infrastructure while maintaining the reliability and performance of IBM Power servers.

Table 6-1 provides an overview of the databases that are available on IBM Power, including supported environments and recommended use cases.

Table 6-1 Database comparisons

Database	Type	Supported OS	Ideal use cases	Power platform support	Notes
IBM Db2 for i	Relational (SQL)	IBM i	Enterprise apps, ERP, OLTP, and RPG/COBOL systems	Native on IBM i	Deep OS integration
IBM Db2 LUW	Relational (SQL)	AIX and Linux	Analytics, data warehousing, and enterprise SQL	Native or container	JSON and REST support
PostgreSQL	Relational (SQL)	Linux	Modern apps, analytics, and microservices	Native or container	Extensible (for example, JSONB and GIS)
MySQL/MariaDB	Relational (SQL)	Linux	Web back ends and reporting	Native or container	Lightweight and widely used
MongoDB	NoSQL (Document)	Linux	Unstructured data, rapid dev, and IoT	Native or container	JSON-like docs and schema-less
Cassandra	NoSQL (Wide Column)	Linux	Big data, time-series, and distributed systems	Native or container	HA
Redis	NoSQL (Key-Value)	Linux	Caching and real-time processing	Native or container	In-memory and fast I/O
CockroachDB	Distributed SQL	Linux (Red Hat OpenShift)	Cloud-native SQL and fault-tolerant apps	Through Red Hat OpenShift	Scales like NoSQL and behaves like SQL.
TimescaleDB	Time-Series (SQL)	Linux (Postgres ext)	Monitoring, IoT, and event data	PostgreSQL-based	Uses Postgres as a back end.
etcd	Key-Value (Config)	Linux (Kubernetes)	Cluster state/config in Kubernetes	Kubernetes env	Used in Red Hat OpenShift and Kubernetes.

For more information about databases, see “Databases” on page 390.

6.4 CI/CD tools

Continuous integration and continuous deployment or delivery (CI/CD) is a foundational DevOps practice that automates the process of building, testing, and deploying software. It enables faster development cycles, improves code quality, and reduces deployment risks.

In the CI phase, developers frequently commit code changes to a shared repository. These commits trigger automated builds and tests to help ensure that new code integrates smoothly with the existing system. This rapid feedback loop helps identify bugs early, encourages collaboration, and maintains high code quality. Popular CI tools include Jenkins, GitHub Actions, and GitLab CI, which all integrate with source control platforms and container environments.

The CD phase automates the delivery or deployment of tested code to production or staging environments. This automation enables organizations to release new features, updates, and fixes more frequently and reliably, minimizing manual intervention and reducing deployment errors.

On IBM Power servers, especially in hybrid environments with IBM i, AIX, and Linux, CI/CD practices support seamless modernization efforts. These pipelines enable integration between earlier systems and modern architectures through microservices, APIs, and containerized workloads. Tools like IBM RDi, Merlin, Red Hat OpenShift, and Ansible can be combined to build robust CI/CD workflows that support agile development while preserving the reliability of core enterprise applications.

6.4.1 GitHub

GitHub is a web-based platform that is built on Git, the open source version control system that enables collaborative software development. It allows multiple developers to work on the same codebase simultaneously, making it a central hub for open source software (OSS) and enterprise development projects.

Here are some of the key features of GitHub:

- ▶ GitHub Copilot is an AI-powered coding assistant that is developed by GitHub and OpenAI. Built on generative AI technology, Copilot understands natural language prompts and generates context-aware code suggestions. Unlike traditional auto-complete tools, Copilot can generate entire lines, functions, or code blocks, accelerating development and reducing boilerplate coding.
- ▶ GitHub supports CI/CD through GitHub Actions, and through integrations with tools like Jenkins and Travis CI.
- ▶ GitHub provides hosting for binary and package builds. Developers can also use GitHub wikis to find instructions about where to download the project for IBM Power.

6.4.2 Red Hat OpenShift Pipelines

Red Hat OpenShift Pipelines is a Kubernetes-native CI/CD solution that is based on the open-source Tekton project. It automates the building, testing, and deployment of applications in containerized environments. Fully integrated into Red Hat OpenShift, it enables developers to define and run CI/CD workflows as Kubernetes resources by using YAML, enabling seamless orchestration alongside application workloads.

Each pipeline step runs in its own container, making pipelines highly scalable, portable, and cloud-native. Developers can build reusable tasks, trigger pipelines through Git events or image changes, and integrate with Git repositories, container registries, and deployment tools.

By adopting Red Hat OpenShift Pipelines, organizations can implement DevOps practices with greater consistency, traceability, and automation across hybrid and multi-cloud environments. This approach empowers development teams to shift left, that is, identifying issues earlier in the lifecycle, and accelerate delivery.

When combined with Red Hat OpenShift GitOps, Red Hat Developer Hub, and IBM DevOps for IBM i, Red Hat OpenShift Pipelines supports a unified CI/CD strategy. This strategy accommodates both modern microservices and earlier workloads running on IBM Power servers, resulting in faster, more reliable software releases with reduced manual intervention and improved collaboration between development and operations teams.

6.4.3 Red Hat OpenShift GitOps

Red Hat OpenShift GitOps is the Red Hat enterprise-grade implementation of GitOps, which is a modern operational model that uses Git as the single source of truth for managing infrastructure and application configurations. Built on the open-source Argo CD project, Red Hat OpenShift GitOps enables automatic synchronization between the state that is defined in Git repositories and the actual state of Red Hat OpenShift or Kubernetes clusters. Any changes to applications, configurations, secrets, or infrastructure are version-controlled, auditable, and automatically deployed through Git commits.

For teams building on IBM Power servers, whether on Linux, IBM i, or AIX (through containerized adapters), Red Hat OpenShift GitOps brings consistency and control across hybrid environments. It supports multi-cluster management, simplifying operations across on-premises, edge, and public cloud deployments.

When combined with tools such as Red Hat OpenShift, Ansible Automation Platform, and IBM Cloud Pak solutions, Red Hat OpenShift GitOps enables a secure, scalable, and automated approach to software delivery. It supports application deployment, infrastructure provisioning, configuration management, and compliance enforcement.

6.5 Support for .NET on IBM Power

Originally developed for the x86 platform, .NET became a cornerstone of modern software development due to its versatility, performance, and rich ecosystem. Here are the key features and benefits of .NET:

- ▶ Developers use .NET to build a wide range of applications, including web, mobile, desktop, cloud-native, gaming, and IoT solutions. Its broad applicability makes it a valuable skill set and a powerful tool for diverse development projects.
- ▶ .NET includes extensive libraries, frameworks, and tools such as ASP.NET Core for web development and Xamarin for mobile applications. The Visual Studio IDE enhances developer productivity with comprehensive features.

- ▶ Through .NET Core and subsequent versions, .NET supports cross-platform development, enabling applications to run on Windows, MacOS, and Linux. This flexibility is essential in today's heterogeneous IT environments.
- ▶ .NET applications benefit from high performance through Just-In-Time (JIT) and Ahead-of-Time (AOT) compilation. The platform also includes robust built-in security features to protect against common vulnerabilities.

The presence of .NET on IBM Power servers is growing, offering new opportunities for application modernization and workload consolidation. Because of open-source nature of .NET Core and collaboration between IBM and Red Hat, .NET Core and later versions are fully supported on the Power Architecture (ppc64le). These versions are available through standard Linux package managers and container registries.

Although Microsoft does not provide direct commercial support for .NET binary files on IBM Power, Red Hat helps ensure timely updates and a stable runtime environment on distributions such as Red Hat Enterprise Linux (RHEL).

The NTi Data Provider is a native .NET solution that enables efficient and secure communication between .NET applications and IBM i servers. Unlike generic drivers, NTi offers optimized performance by directly interacting with IBM i resources such as Db2 databases, Control Language (CL) commands, and programs. Its native .NET architecture helps ensure cross-platform compatibility and eliminates the need for special IBM i installations.

By simplifying integration and using secure internal networks on IBM Power servers, NTi enables .NET developers to extend existing IBM i business logic with modern .NET capabilities, which results in a faster, more secure, and developer-friendly modernization workflows.

6.6 Kubernetes

Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. Originally developed by Google and now maintained by the Cloud Native Computing Foundation (CNCF), Kubernetes simplifies the complexities of running large-scale applications by providing a robust framework for container management. It allows developers and IT teams to manage applications across clusters of machines, helping ensure HA, scalability, and fault tolerance.

Here are the key features of Kubernetes:

- ▶ Kubernetes abstracts the underlying infrastructure, enabling applications to be deployed and managed consistently across different environments, whether on-premises or in the cloud.
- ▶ Kubernetes automates many operational tasks, including load-balancing, service discovery, and self-healing, where it can automatically restart or reschedule failed containers, which makes it ideal for microservices architectures, where applications are broken down into smaller, more manageable services that run independently but work together.

- ▶ Kubernetes uses a declarative approach to configuration, where the state of the system is defined in configuration files (typically YAML or JSON). These files specify the containers to run, their resources, networking, and other parameters, and Kubernetes helps ensure that the system matches this state.
- ▶ Kubernetes supports features such as rolling updates, versioned deployments, and persistent storage management, making it an essential tool for modern DevOps practices and continuous delivery pipelines.

Kubernetes became the standard for container orchestration and is widely adopted across industries for building and managing cloud-native applications.

For more information about Kubernetes, see 10.2.2, “Kubernetes” on page 353.

6.6.1 Red Hat OpenShift

Red Hat OpenShift is a leading hybrid cloud application platform that streamlines the entire application lifecycle. It provides a comprehensive set of tools and services for building, deploying, and managing applications consistently across any infrastructure, with full-stack automated operations.

Powered by Kubernetes, Red Hat OpenShift combines container orchestration with production-grade workloads by using Docker container management tools. The platform offers a secure, isolated, and self-healing environment for managing containerized applications. Its portability and extensibility make it ideal for organizations adopting microservices and cloud-native architectures.

Red Hat OpenShift enables organizations to accelerate their cloud-native journey while using the performance, reliability, and scalability of IBM Power servers. Available in both self-managed and fully managed cloud service editions, Red Hat OpenShift supports a wide range of workloads, including cloud-native, AI, virtualized, and traditional applications.

In February 2025, Red Hat OpenShift 4.18 introduced several enhancements that support modernization efforts on IBM Power servers:

- ▶ Multiple Architecture Cluster support.
- ▶ Multiarch Tuning Operator.
- ▶ Secondary Scheduler Operator.
- ▶ Tuning etcd latency tolerances.
- ▶ Installer Provisioned Infrastructure (IPI) for IBM PowerVS through CAPI.
- ▶ The Red Hat OpenShift command-line interface (CLI) (`oc`) now can add compute nodes to on-premises clusters.

For more information about Red Hat OpenShift, see 10.2.1, “Red Hat OpenShift” on page 351.

6.6.2 IBM Cloud Pak

IBM Cloud Pak is a suite of pre-integrated, containerized software solutions that run on Red Hat OpenShift. They help businesses modernize applications, automate operations, manage data, and implement AI across hybrid cloud environments. Each IBM Cloud Pak includes IBM middleware, open-source components, Kubernetes operators, and certified security. These elements provide enterprises with a consistent and scalable foundation for digital innovation.

There are several IBM Cloud Pak offerings, each focused on a specific domain:

- ▶ IBM Cloud Pak for Applications
- ▶ IBM Cloud Pak for Data (CP4D)
- ▶ IBM Cloud Pak for Integration (CP4I)
- ▶ IBM Cloud Pak for Business Automation (CP4BA)
- ▶ IBM Cloud Pak for AIOps

IBM Cloud Pak for Applications

IBM Cloud Pak for Applications is an enterprise-ready, containerized software solution that helps modernize existing applications and develop new cloud-native apps. Built on IBM WebSphere® offerings and Red Hat OpenShift Container Platform, it provides a comprehensive set of tools to support migrations across public, private, and hybrid cloud environments.

IBM Cloud Pak for Applications includes IBM Cloud Transformation Advisor, which is an AI-powered tool that helps re-factor and re-architect applications. The solution provides automated vulnerability assessment and identification to help ensure continuous security compliance across all deployment environments. It also automates audit reporting to simplify compliance management. Developers can use their preferred IDEs to build and deploy applications, with support for modern run times and DevOps workflows. This integration streamlines development and enhances productivity.

IBM Cloud Pak for Data

CP4D is a unified, pre-integrated data and AI platform that helps organizations collect, organize, analyze, and infuse AI into their data. It simplifies data management and supports AI and analytics workloads. Running natively on the Red Hat OpenShift Container Platform, CP4D supports deployment across multiple cloud environments, including IBM Cloud, Amazon Web Services (AWS), and Microsoft Azure.

The platform provides secure access to data at its source, eliminating the need for data migration and reducing data silos. It helps ensure seamless data integration and creates a trusted, business-ready analytics foundation. CP4D simplifies data preparation, policy enforcement, security, and compliance while automating data governance and ML learning models, and scales these capabilities consistently across the organization to support comprehensive data analysis and insights.

By integrating AI across the business with trust and transparency, the platform supports the end-to-end AI workflow and enables effective integration of AI into business processes. CP4D offers a single interface for end-to-end analytics with built-in governance, simplifying the management of data and AI capabilities. Its scalable Kubernetes environment enables organizations to expand their data and AI operations as needed. Supporting multi-cloud deployments, the platform provides agility and avoids vendor lock-in, helping accelerate the journey to AI and unlock the value of data for AI-driven digital transformation.

IBM Cloud Pak for Integration

CP4I is a comprehensive, AI-powered hybrid integration platform that connects applications, data, systems, and services across any environment. It provides a unified experience through a suite of integration tools that streamlines the creation, management, and deployment of integrations. Running on Red Hat OpenShift, CP4I supports both cloud and on-premises deployments, helping ensure scalability and security. The platform includes components such as IBM API Connect® for API management, IBM App Connect for no-code integration, and IBM Event Streams for real-time data processing. By using AI and automation, CP4I accelerates integration, reduces manual workflows, and improves responsiveness to real-time events. These capabilities make it an ideal solution for organizations modernizing their integration strategies and advancing digital transformation.

IBM Cloud Pak for Business Automation

CP4BA is a modular set of integrated software components that automate work and accelerate business growth. Built for hybrid cloud environments, it simplifies complex workflows, supports records management, and improves overall productivity. The platform uses AI to identify gaps and create low-code and no-code automations, helping streamline operations. Running on Red Hat OpenShift, CP4BA supports containerized deployments across multiple cloud environments, offering flexibility and scalability. Key features include automating case and process workflows, converting unstructured content into actionable data, and using software robots to complete tasks based on AI insights. This comprehensive automation solution helps organizations increase efficiency, reduce operational costs, and drive continuous process improvement.

IBM Cloud Pak for AIOps

IBM Cloud Pak for AIOps is an advanced, AI-powered platform that enhances IT operations (ITOps) by using AI and ML. It integrates with existing ITOps toolchains to provide comprehensive visibility, proactive incident management, and automated remediation. By analyzing data from sources such as logs, metrics, and events, IBM Cloud Pak for AIOps helps IT teams predict and resolve issues before they affect business operations. The platform supports hybrid cloud environments, enabling organizations to manage IT infrastructure across on-premises, cloud, and containerized environments. Key features include event correlation and compression, anomaly detection, root cause analysis, and automated runbooks. These capabilities reduce mean time to resolution (MTTR) and improve operational efficiency. With collaborative tools and real-time insights, IBM Cloud Pak for AIOps empowers IT teams to innovate faster, reduce operational costs, and help ensure the reliability of mission-critical workloads.



Tools and performance

Modern application environments, which are defined by cloud-native architectures, microservices, containerization, and DevOps practices, introduce distinct challenges for tools and performance management. These environments are dynamic, distributed, and complex, which renders traditional monitoring and management approaches ineffective.

Modern applications often rely on microservices that run across multiple servers or cloud environments. This distribution increases complexity because tools must track interactions among services that might use different languages and technologies. Tools must also accommodate the dynamic nature of these applications, where services scale up or down, and the infrastructure continuously changes.

Traditional performance monitoring often focuses on isolated components of the stack, such as CPU usage, memory consumption, or database performance. However, modern applications require an end-to-end view that spans front-end performance and back-end infrastructure. Achieving this visibility across distributed environments is challenging.

Modernized applications generate vast volumes of performance data, including logs, metrics, traces, and events. Collecting, storing, and analyzing this data can overwhelm monitoring systems and hinder their ability to extract meaningful insights. Tools must distinguish between normal behavior and actual issues. Filtering out irrelevant or redundant data is essential, but improper filtering can result in missed signals.

Many modern applications use auto-scaling to adjust resource allocation dynamically based on demand. Tools must manage and predict resource consumption effectively to prevent issues such as resource contention or over-provisioning. As applications scale, so do costs, especially in cloud environments. Managing resource usage and cost becomes a balancing act: under-utilization leads to inefficiencies, and over-provisioning results in unnecessary expenses.

As artificial intelligence (AI) and machine learning (ML) play a growing role in application management, tools must predict potential performance degradation before it occurs. This capability requires access to large datasets and the ability to process them efficiently, along with confidence in the accuracy of the algorithms. There is a growing shift toward autonomous performance management systems that can detect and resolve issues automatically. However, these systems are still evolving and might not always respond correctly in complex or unexpected scenarios.

The challenges of tools and performance management in modern application environments stem from the complexity, scale, and rapid pace of change in today's software ecosystems. To address these challenges, organizations need a cohesive strategy for selecting, integrating, and optimizing performance management tools. This strategy should emphasize automation, real-time monitoring, and cross-team collaboration to proactively manage performance and maintain a smooth, responsive user experience.

The following topics are described in this chapter:

- ▶ 7.1, “Monitoring” on page 208
- ▶ 7.2, “Logging” on page 220
- ▶ 7.3, “Autoscaling and quality of service” on page 225
- ▶ 7.4, “Management” on page 226

7.1 Monitoring

Monitoring is foundational to any successful application modernization effort. Before refactoring code or deploying new services, teams must establish a robust monitoring framework. This initial phase involves capturing the current state of application performance. Key metrics such as response times for critical user flows, error rates across components, infrastructure resource usage (CPU, memory, and network bandwidth), and user experience indicators serve as essential benchmarks. These baselines provide an objective view of the application’s initial state, enabling teams to set measurable goals for improvement. Without this visibility, it becomes difficult to assess the impact of modernization or detect regressions.

As the modernization journey progresses, whether through incremental refactoring, re-platforming to cloud environments, or a complete re-architecture by using microservices, continuous monitoring serves as a critical guide. It provides real-time visibility into application health and performance at every stage of the transformation. Deploying a new service version, migrating a database, or integrating a new application programming interface (API) can introduce unexpected issues. Monitoring tools help detect deviations from expected behavior, such as increased latency, elevated error rates, or resource contention. This early detection enables development and operations teams to address problems proactively, preventing minor issues from escalating into major outages or performance degradation. Monitoring also validates each deployment, confirming that modernized components function correctly in the new environment and integrate seamlessly with existing systems.

Once the application is fully modernized and deployed in its target environment, monitoring remains essential, although its focus evolves. The new architecture, often based on distributed systems and cloud-native technologies, introduces additional complexity. Monitoring now targets performance bottlenecks that might arise in this environment, such as inefficient inter-service communication, suboptimal database queries, or cloud resource limitations. By continuously analyzing resource usage patterns, teams can optimize allocation and reduce costs in cloud environments. Monitoring also supports scalability by tracking performance under varying load conditions, helping teams identify areas that require further optimization to maintain responsiveness and stability.

Beyond performance, monitoring is essential for maintaining the reliability and stability of modernized applications. It continuously tracks uptime and availability to help ensure compliance with defined service-level agreements (SLAs). Comprehensive error tracking and logging provide the detailed data that is needed to diagnose root causes and resolve issues quickly, minimizing downtime. Proactive alerting, which is configured around key performance indicators (KPIs), acts as an early warning system, notifying teams of potential issues before they escalate. This approach shifts the focus from reactive incident management to preventive action, enhancing the overall resilience of the application.

A user-centric approach to modern application development requires a clear understanding of the actual user experience. Real User Monitoring (RUM) tools, which are integrated into the monitoring strategy, provide valuable insights into how users interact with the application. Metrics such as page load times, JavaScript errors, and perceived responsiveness are captured and analyzed. This data helps identify areas where users might encounter friction, guiding optimization efforts to improve satisfaction and enhance the overall user journey.

Monitoring provides the data-driven foundation for continuous improvement and iterative development. The performance, reliability, and user experience data that it collects informs future development decisions. After implementing enhancements or introducing new features, monitoring validates their effectiveness and identifies any unintended consequences. This continuous feedback loop helps ensure that the modernized application remains performant, reliable, and aligned with evolving user needs and business goals. Monitoring is both a technical requirement and a strategic imperative for realizing the full value of application modernization.

7.1.1 Monitoring solutions

Monitoring solutions are integral to modern infrastructure, offering real-time insights and proactive management capabilities that support optimal performance, security, and reliability. This section highlights a selection of available solutions:

- ▶ Instana is a robust observability platform that uses AI and automation to monitor applications across cloud-native environments. It delivers high-fidelity data with updates every second, enabling rapid identification and resolution of issues before they affect users. Instana provides comprehensive monitoring across the entire application stack, from microservices to Kubernetes, which provides a unified view of system performance.
- ▶ Turbonomic is an advanced monitoring solution that is focused on Application Resource Management (ARM). It dynamically allocates resources in real time to optimize performance and reduce costs. With intelligent automation and proactive resource management, Turbonomic helps ensure continuous application performance across hybrid and multicloud environments.
- ▶ The Red Hat OpenShift monitoring stack integrates tools such as Prometheus, Grafana, and Alertmanager to deliver detailed metrics, visualization, and alerting for containerized applications. This stack monitors both platform components and user workloads, supporting the health and efficiency of modern IT environments.

IBM Instana Observability

IBM Instana Observability is a comprehensive, fully automated enterprise observability platform that is designed for modern, dynamic applications, including ones that are built with microservices, containers (such as Kubernetes and Docker), and cloud-native architectures. It provides contextual insights that enable teams to take intelligent action and help ensure optimal application performance.

A key differentiator of Instana is its automatic discovery and monitoring of the entire application stack and underlying infrastructure. After the Instana agent is deployed, it automatically detects and begins monitoring more than 300 technologies without requiring extensive manual configuration. The technologies include application run times (such as Java, Node.js, and Python), databases, middleware, and orchestration platforms. Instana dynamically builds a real-time dependency map that visualizes the relationships among services and infrastructure components.

Core capabilities of IBM Instana Observability include full-stack visibility, which offers insights from user interactions down to the code and infrastructure levels. It captures 100% of requests and traces at a 1-second granularity without sampling, eliminating blind spots in performance monitoring. The platform uses AI-powered analytics for automatic root cause analysis, helping teams quickly identify the source of performance issues and reduce mean time to resolution (MTTR). Other features include anomaly detection for proactive issue identification, end-user experience monitoring (EUM), infrastructure monitoring, contextual log management, synthetic monitoring for proactive testing, and customizable dashboards for visualizing key metrics. IBM also offers Instana Observability on z/OS, extending these capabilities to mainframe environments and enabling end-to-end visibility in hybrid applications.

IBM Instana Observability offers multiple pricing tiers, including Essentials (focused on infrastructure monitoring) and Standard (providing full-stack observability with application performance monitoring, tracing, and more). Pricing is typically based on the number of Managed Virtual Servers (MVSs) or hosts being monitored, and the selected features. Deployment options include software as a service (SaaS) and self-hosted environments, with self-hosted generally incurring higher costs. Extra charges might apply for features such as synthetic monitoring from IBM managed Points of Presence (PoPs) and extended log retention. IBM provides a pricing calculator for estimates, and private offers or no-charge trials might be available through its website or cloud marketplaces.

For more information about Instana, see [IBM Instana Observability](#).

IBM Turbonomic

IBM Turbonomic is an ARM platform that uses AI to continuously optimize application performance and cost efficiency across hybrid and multicloud environments. Its primary goal is to help ensure that applications consistently receive the precise resources that they need to operate effectively while minimizing infrastructure costs. Turbonomic achieves this goal through dynamic, real-time matching of application demand with the available infrastructure supply.

At the core of the Turbonomic operation is its unique approach to modeling the IT environment as a dynamic marketplace. In this model, applications act as consumers requesting resources such as CPU, memory, storage, and network bandwidth, and the infrastructure serves as the provider. Turbonomic uses a sophisticated economic scheduling engine to analyze real-time resource consumption and associated costs, which are represented in a virtual currency. This continuous analysis enables the platform to make intelligent, automated decisions about resource allocation, instance rightsizing, and workload placement, which strikes a balance between application performance and infrastructure cost.

Turbonomic offers extensive capabilities that are designed to provide comprehensive control over resource management. Full-stack visibility is a core feature, with the platform automatically discovering and mapping relationships across the application and infrastructure stack, from the application layer to physical hardware. This holistic view provides an essential context for identifying resource dependencies and potential bottlenecks. Turbonomic also uses AI and ML to analyze large volumes of real-time data and predict future resource needs. This proactive approach enables the platform to detect performance risks and cost inefficiencies before they affect application availability or budgets.

A key strength of Turbonomic is its ability to automate resource optimization. The platform generates actionable recommendations based on real-time analysis and can be configured to run these actions automatically. These actions include dynamically resizing virtual machines (VMs), migrating workloads across hosts or cloud regions, adjusting cloud instance types to match demand, and scaling Kubernetes resources as needed. Automation reduces manual intervention and helps ensure continuous optimization. Turbonomic also supports cloud cost optimization by identifying over-provisioned resources by using reserved instances and savings plans, and recommending cost-effective instance types for specific workloads.

Turbonomic also plays a critical role in performance assurance. By continuously monitoring application resource consumption and dynamically adjusting resource allocation based on real-time demand, the platform helps ensure that applications consistently meet their defined service-level objectives (SLOs). Its support for hybrid and multicloud environments is a key advantage, enabling seamless resource management across on-premises data centers, including technologies such as VMware and hyperconverged infrastructure, and major public cloud providers such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud. For organizations adopting containerization, Turbonomic offers specialized Kubernetes optimization capabilities, including intelligent container resizing, optimal pod placement within clusters, and dynamic cluster scaling to improve performance and reduce costs in complex environments.

Beyond real-time management, Turbonomic offers powerful what-if planning capabilities. Users can simulate scenarios such as capacity planning, cloud migration strategies, and infrastructure upgrades to assess their potential impact on application performance and infrastructure costs before making changes. Turbonomic also features a robust integration ecosystem, connecting seamlessly with a wide range of IT management and DevOps tools, including monitoring solutions, cloud management platforms, and automation frameworks to deliver a unified and streamlined operational experience.

For more information about Turbonomic, see the [IBM Turbonomic Application Resource Management™ documentation](#).

Red Hat OpenShift monitoring stack

Red Hat OpenShift includes a robust and comprehensive monitoring stack that is designed to maintain the health and performance of containerized applications and infrastructure. This stack integrates several key tools and components that work together to deliver detailed insights and proactive management capabilities.

Figure 7-1 shows a high-level overview of the monitoring stack.

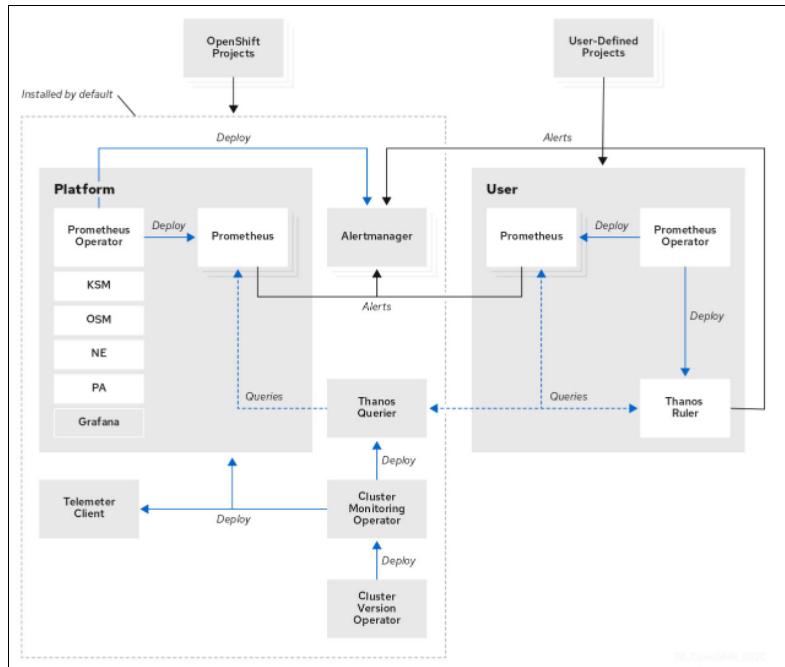


Figure 7-1 Overview of the Red Hat OpenShift monitoring stack

Prometheus

Prometheus is a core component of the Red Hat OpenShift monitoring stack. This open-source system monitoring and alerting toolkit collects and stores metrics as time series data. Prometheus is highly scalable and capable of handling millions of time series, making it suitable for large, complex environments. It scrapes data from various endpoints and stores it in its own time series database, providing detailed insights into system performance.

Grafana

Grafana works with Prometheus to visualize collected metrics. This open-source platform enables users to create and share dashboards that provide a comprehensive view of system health and performance. Grafana supports a wide range of data sources and offers powerful querying capabilities, making it simple to analyze and interpret the data that is collected by Prometheus.

Alertmanager

Alertmanager is a critical component of the Red Hat OpenShift monitoring stack. It processes alerts that are generated by Prometheus and manages their delivery. Alertmanager can route alerts to various notification channels, such as email, Slack, or PagerDuty to help ensure that the appropriate personnel are promptly informed of issues. It also supports alert deduplication, grouping, and silencing to reduce alert fatigue and improve response times.

Sysdig

Sysdig is an additional monitoring tool that integrates with Red Hat OpenShift. It provides deep visibility into containerized environments and offers features such as container security, compliance, and performance monitoring. Sysdig uses a kernel module to capture system calls and other operating system (OS)-level events, delivering detailed insights into container activity and performance.

Datadog

Datadog is a cloud-based monitoring and analytics platform that integrates with Red Hat OpenShift to monitor applications, infrastructure, and logs. It provides real-time visibility into system performance and integrates with a wide range of technologies, making it a versatile tool for monitoring modern IT environments. Datadog offers features such as anomaly detection, dashboards, and alerting to help organizations maintain optimal system health.

Red Hat OpenShift Web Console

Red Hat OpenShift Web Console includes built-in monitoring features that provide visual representations of cluster metrics. It offers default dashboards that help administrators quickly assess the state of the cluster. The console also includes tools for managing metrics, alerts, and monitoring dashboards, making it simpler to monitor and manage the Red Hat OpenShift environment.

User-defined monitoring

Red Hat OpenShift enables cluster administrators to enable monitoring for user-defined projects. This feature provides more monitoring components that can be configured to monitor specific services and pods within user projects. It helps ensure that both platform components and user workloads are monitored effectively, providing comprehensive coverage across the environment.

In summary, the Red Hat OpenShift monitoring stack is designed to deliver detailed insights into system performance, security, and reliability. By using tools such as Prometheus, Grafana, Alertmanager, Sysdig, and Datadog, organizations can ensure that their containerized applications and infrastructure operate efficiently. The built-in monitoring features of the Red Hat OpenShift web console further enhance visibility and control, making it simpler to manage and optimize the environment.

7.1.2 Using Prometheus and Grafana on Red Hat OpenShift with IBM Power

This section explains how to scrape Prometheus metrics from a Go application that is instrumented for Prometheus data. The application is deployed on a Red Hat OpenShift cluster running on IBM Power, along with Prometheus and Grafana.

Running a Prometheus instrumented application on Red Hat OpenShift

This section describes how to run a Prometheus instrumented application on Red Hat OpenShift by using a Go application that is instrumented for Prometheus. For more information about this application, see this [Prometheus documentation](#).

Complete the following steps:

1. Deploy the Dockerfile that you find at this [GitHub repository](#). Use this file to create and push an image to a Quay repository, as shown in Example 7-1.

Example 7-1 Deploying the application

```
#podman build . -t quay.io/mithunibm87/metrics:latest  
#podman push quay.io/mithunibm87/metrics:latest
```

Figure 7-2 shows the resulting deployment.

The screenshot shows the Red Hat OpenShift web interface. The left sidebar is titled "Administrator" and includes links for Home, Operators, Workloads (Pods, Deployments, DeploymentConfigs, StatefulSets, Secrets, ConfigMaps), CronJobs, Jobs, DaemonSets, and ReplicaSets. The "Deployments" link is currently selected. The main content area is titled "Project: metrics-demo" and shows "Deployment details" for a deployment named "metrics". The deployment has 1 Pod. It is running in the "metrics-demo" Namespace. The "Update strategy" is set to "RollingUpdate". The "Max unavailable" setting is "25% of 1 pod". There is an "Edit" button and a "Max surge" setting of "25% greater than 1 pod".

Figure 7-2 Project deployment details

2. Deploy the Prometheus instance by using a publicly available Docker image that is available for Linux (ppc64le) at docker.io by running the following command:

```
#docker pull prom/Prometheus
```

Figure 7-3 shows the Prometheus deployment.

The screenshot shows the Red Hat OpenShift web interface. The left sidebar is titled 'Administrator' and includes links for Home, Operators, Workloads (Pods, Deployments, DeploymentConfigs, StatefulSets, Secrets, ConfigMaps), CronJobs, Jobs, DaemonSets, and ReplicaSets. The 'Deployments' link is currently selected. The main content area shows a deployment named 'prometheus' in the 'metrics-demo' project. The 'Deployment details' tab is selected. It displays a summary: 1 Pod. Below this, there are fields for Name (prometheus), Namespace (NS metrics-demo), and Labels. On the right, there are sections for 'Update strategy' (RollingUpdate), 'Max unavailable' (25% of 1 pod), and 'Max surge' (25% greater than 1 pod). There is also an 'Edit' button.

Figure 7-3 Prometheus deployment

3. Create a `prometheus.yml` configuration map that contains the scrape information. The default Prometheus setup does not automatically know where to scrape data from for the instrumented application. Therefore, you create a configuration map that includes these details, as shown in Example 7-2.

Example 7-2 The `prometheus.yml` file

```
global:  
  scrape_interval:      15s # Set the scrape interval to every 15 seconds. The  
  # default is every 1 minute.  
  evaluation_interval: 15s # Evaluate the rules every 15 seconds. The default is  
  # every 1 minute.  
  alerting:  
    alertmanagers:  
      - static_configs:  
        - targets:  
          # - alertmanager:9093  
    # Load rules once and periodically evaluate them according to the global  
    # 'evaluation_interval'.  
    rule_files:  
    scrape_configs:  
      - job_name: 'mv-metrics'  
        scrape_interval: 5s  
        metrics_path: /metrics  
        static_configs:  
          - targets: ['metrics.metrics-demo.svc.cluster.local:9080']
```

The results should look like Figure 7-4.

```
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
alerting:
  alertmanagers:
    - static_configs:
      - targets:
          # - alertmanager:9093
# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  scrape_configs:
    - job_name: 'mv-metrics'
      scrape_interval: 5s
      metrics_path: /metrics
      static_configs:
        - targets: ['metrics.metrics-demo.svc.cluster.local:9080']
```

Figure 7-4 ConfigMap for Prometheus

This file can be found at [this GitHub repository](#).

4. In the default Prometheus instance, create volumes and volume mounts. Then, load the `prometheus.yml` file and force the instance to read the updated `prometheus.yml` file by completing the following steps:
 - a. Under `volumes` in the Prometheus deployment file, add the lines that are shown in Example 7-3.

Example 7-3 Adding volumes

```
volumes:
  - name: prometheus-config
    configMap:
      name: prometheus-config
      items:
        - key: prometheus.yml
          path: prometheus.yml
          mode: 420
    defaultMode: 420
```

- b. Under `volumeMount` in the Prometheus deployment file, add the lines that are shown in Example 7-4.

Example 7-4 VolumeMount definition

```
volumeMounts:
  - name: prometheus-config
    mountPath: /etc/prometheus/prometheus.yml
    subPath: prometheus.yml
```

- c. Pass the arguments that are shown in Example 7-5 to the image to read the new Prometheus file.

Example 7-5 Arguments for Prometheus

args:

- `--config.file=/etc/prometheus/prometheus.yml'`
 - `--storage.tsdb.path=/prometheus'`
 - `--web.console.libraries=/usr/share/prometheus/console_libraries'`
 - `--web.console.templates=/usr/share/prometheus/consoles'`
-

To find the complete modified Prometheus deployment, see [this GitHub repository](#).

5. Check for targets on Prometheus. You see that Prometheus displays the target for the instrumented Go application, as shown in Figure 7-5.

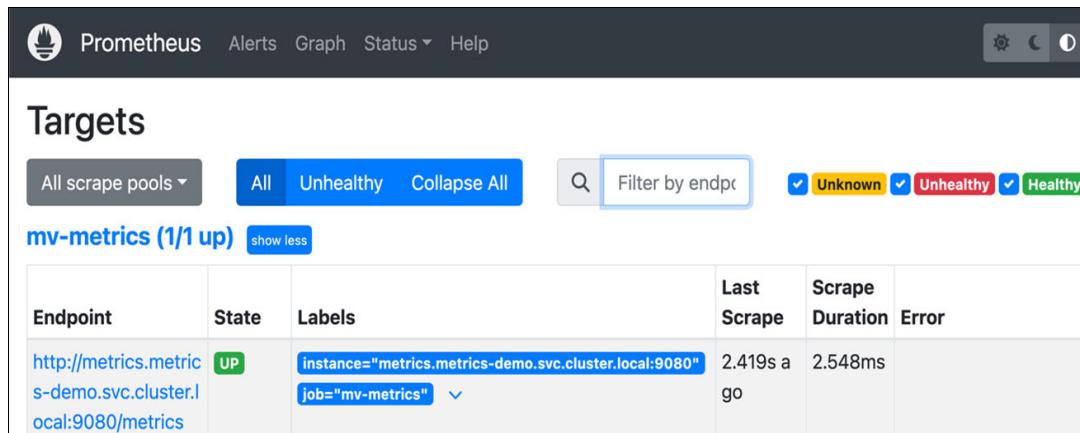


Figure 7-5 Prometheus instance after configuration

6. To check the number of total operations that were processed, see the `myapp_processed_ops_total` parameter, as shown in Figure 7-6.

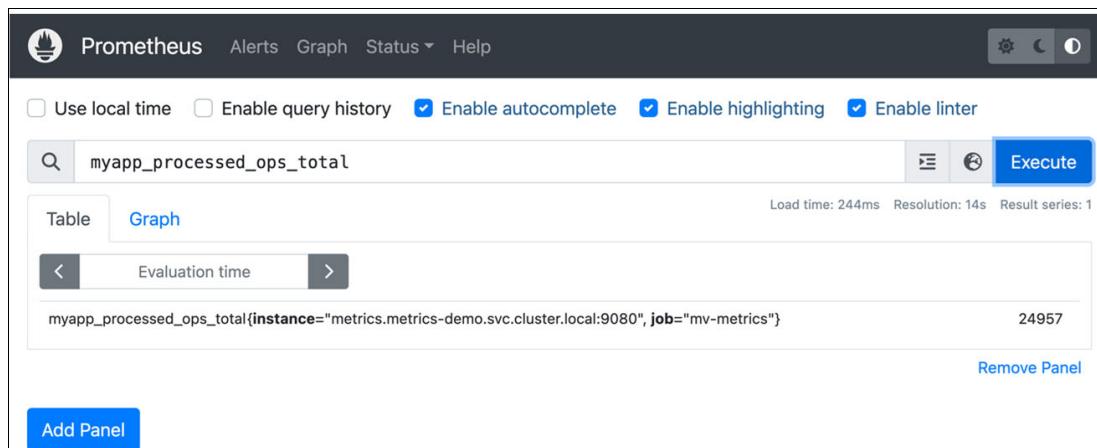


Figure 7-6 Operations processed

7. Deploy Grafana by completing the following steps. To find the Grafana image for Linux (ppc64le), see the [Red Hat catalog](#).

- a. Deploy the Grafana image on Red Hat OpenShift, as shown in Figure 7-7.

The screenshot shows the Red Hat OpenShift web interface. The left sidebar is titled "Administrator" and includes sections for Home, Operators, Workloads (Pods, Deployments, DeploymentConfigs, StatefulSets, Secrets, ConfigMaps, CronJobs, Jobs, DaemonSets, ReplicaSets), and Metrics. The "Deployments" section is currently selected. The main content area shows a deployment named "grafana" in the "metrics-demo" project. The "Deployment details" tab is selected. It displays a summary of the deployment: 1 Pod, Name: grafana, Namespace: NS metrics-demo, and Labels: ann.grafana, ann.kubernetes.io/component=grafana, ann.kubernetes.io/instance=grafana. The "Update strategy" is set to RollingUpdate, "Max unavailable" to 25% of 1 pod, and "Max surge" to 25% greater than 1 pod. A blue circular icon indicates the status of the single pod.

Figure 7-7 Grafana deployment

- b. Within Grafana, to add a data source, select **Data sources**, as shown in Figure 7-8.

The screenshot shows the Grafana dashboard interface. The left sidebar includes options for Home, Search dashboards, Starred, Dashboards (Browse, Playlists, Snapshots, Library panels, + New dashboard, + New folder, + Import), Explore, Alerting, Configuration, and Data sources. The "Data sources" option is highlighted with a dark grey background. The main content area features a "Need help?" section with links to documentation and a "COMPLETE" section prompting to "Add your first data source". There are also links to "Learn how in the docs" and a "Latency" chart at the bottom right.

Figure 7-8 Selecting Data sources

- c. Provide the URL of the Prometheus instance that you deployed, as shown in Figure 7-9 on page 219.

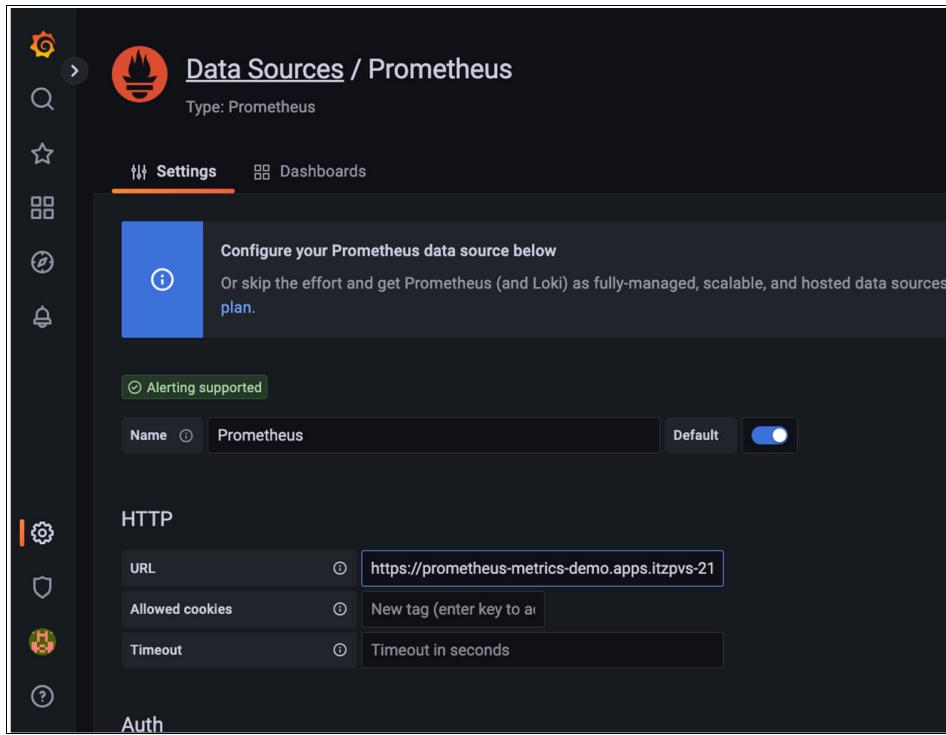


Figure 7-9 Entering the URL of the Prometheus instance

- d. Select **Save & test**. If you are successful, you see the message “Data source is working”, as shown in Figure 7-10.

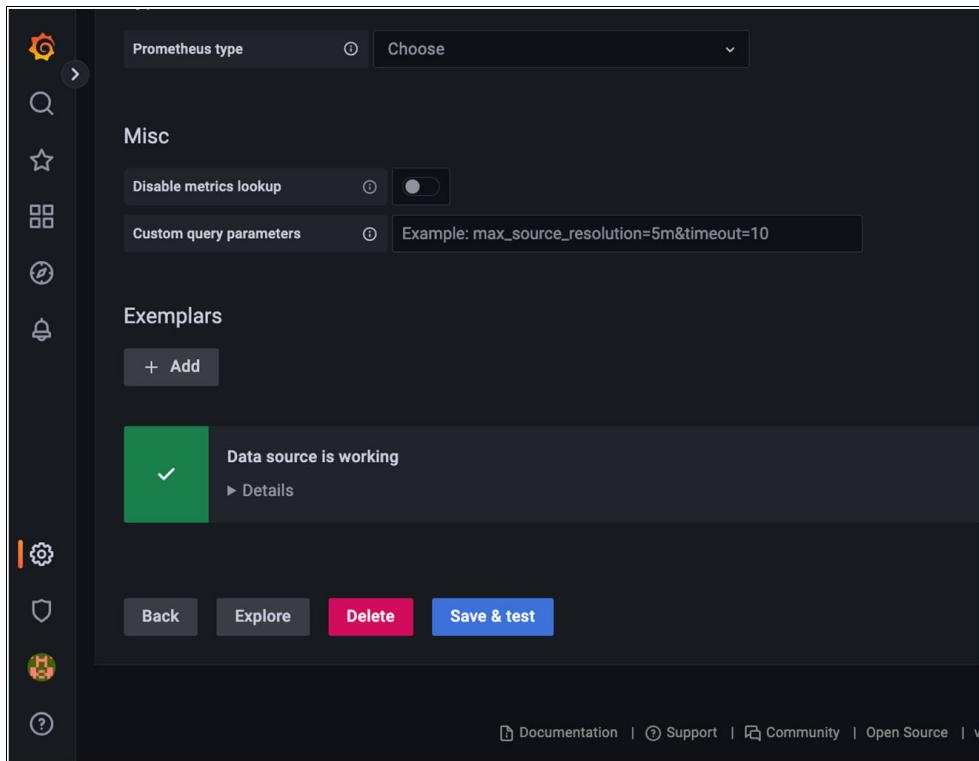


Figure 7-10 Successful data connection

- Create a dashboard and add the new panel with data source Prometheus. You can then query the data. Figure 7-11 shows myapp_processed_ops_total for the past hour.

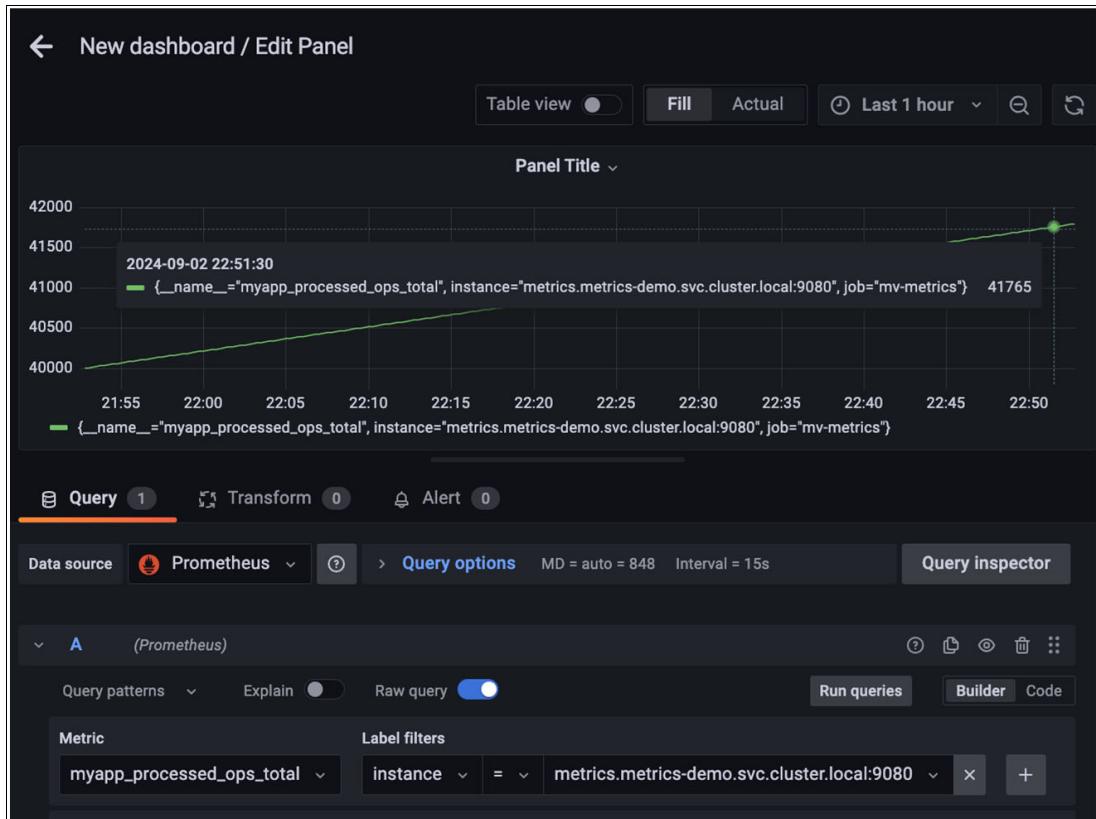


Figure 7-11 Dashboard showing statistics

This section demonstrated how to create a complete monitoring stack by using a customized Prometheus configuration and deploying Grafana on a Red Hat OpenShift cluster running on IBM Power.

7.2 Logging

Logging involves capturing events such as user and application actions, system and server events, transactions, and errors. It can occur at various levels, including system, server, network, and application. Logged events are often categorized by severity levels, such as trace, debug, info, warning, error, and fatal.

Integrating a logging framework during the early stages of application design is essential. Application logging helps track how the application behaves under different conditions and workloads over time. By logging contextual information about the specific module, code block, or component responsible for an event, developers can better understand application behavior and effectively debug errors.

This section describes the most commonly used logging software, components, and capabilities in modern environments.

7.2.1 Traditional system logging

AIX uses the syslog daemon (`syslogd`) to filter and log system events. These events can be redirected to a central logging server. In Red Hat Enterprise Linux, this daemon is called `rsyslogd`. Administrators can typically identify messages originating from a specific virtual machine (VM) that hosts critical applications.

In a modern Red Hat OpenShift environment running on IBM Power, managing, redirecting, and filtering logging events across a multi-node, multi-container cluster involves a layered and integrated approach.

7.2.2 Container logging

Containers typically log to standard output streams, such as `stdout` and `stderr`. However, the log format varies by container run time. For Docker, the default format is JSON. For CRI-O, which Red Hat OpenShift uses, the format is plain text. As Red Hat OpenShift migrates from Docker to CRI-O as its default container engine, not all logging tools can accurately parse and format all container logs.

Container logging is illustrated in Figure 7-12.

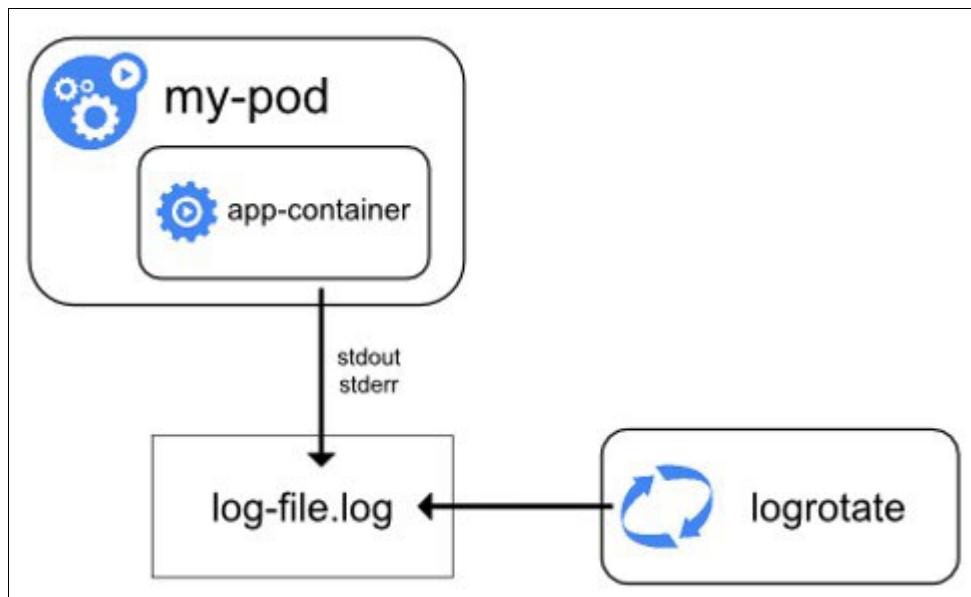


Figure 7-12 Kubernetes logging¹

A container run time manages and redirects output from a containerized application's `stdout` and `stderr` streams. Although container run times implement this behavior differently, their integration with the kubelet follows the standardized CRI logging format.

When a container restarts, Kubernetes retains one terminated container and its logs. If a pod is evicted from a node and then removed or deleted, all associated containers and their logs are also removed. The kubelet exposes these logs to clients through a feature of the Kubernetes API, which is typically accessed by using the `kubectl logs` or `oc logs` command.

¹ Source: <https://kubernetes.io/docs/concepts/cluster-administration/logging>

The challenge is to persist log data, which is similar to traditional syslogs, by redirecting and filtering logs to a central log server. In a modern infrastructure, this server is often implemented as a central logging container.

7.2.3 Red Hat OpenShift Logging

Red Hat OpenShift Logging enables *observability*. Because the architecture is declarative, it enables observation of system behavior.

You can install Red Hat OpenShift Logging as an operator in Red Hat OpenShift on IBM Power, as shown in Figure 7-13.

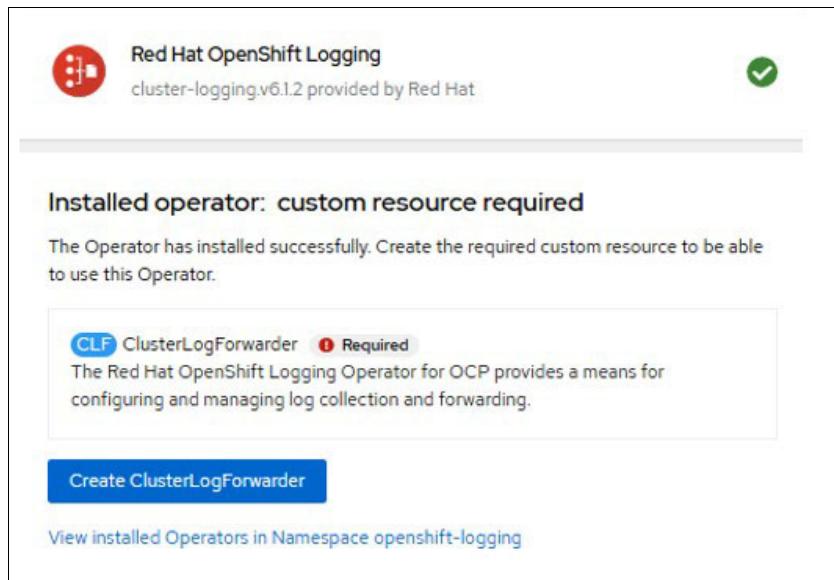


Figure 7-13 Logging operator

Since Version 5.6, Red Hat OpenShift Logging includes stream-based, retention capabilities within the Loki Stack. This feature enables persistent storage of container logs in an object storage bucket, helping ensure that log data is retained even after the associated container or pod is removed, as shown in Figure 7-14 on page 223.

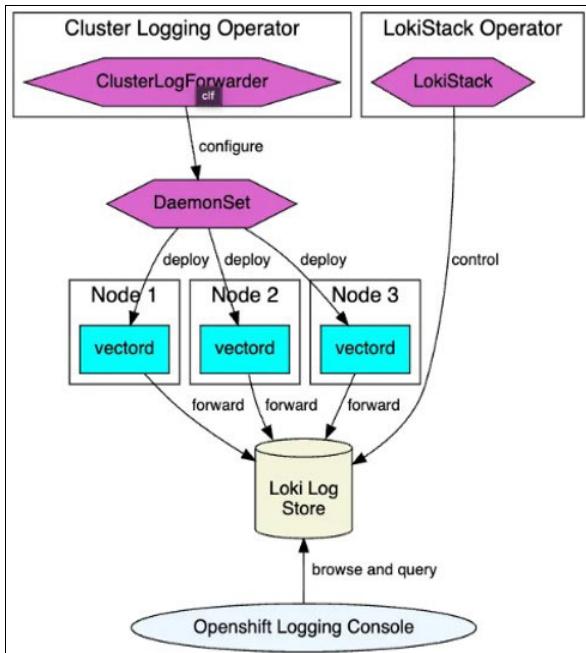


Figure 7-14 Integrating the logging operator

The vector log collector filters and collects the log stream. It forwards the required log data to persistent, S3-compliant object storage in the Loki log store.

All log events are accessible through a console plug-in for observability. You can also configure Red Hat OpenShift Logging to forward logs to your log server.

For more information about the Red Hat OpenShift Logging stack, see this [Red Hat document](#).

Alternatively, you can configure only the collector in Red Hat OpenShift Logging to forward logs to a third-party log aggregator for storage and analysis. This approach reduces the load and resource requirements on the Red Hat OpenShift cluster.

Here are examples of third-party logging servers that can receive logs from Red Hat OpenShift Logging:

- ▶ Elasticsearch
- ▶ Grafana Loki
- ▶ Splunk
- ▶ Amazon CloudWatch
- ▶ Google Cloud Logging

7.2.4 Apache Log4j

Apache Log4j is a widely used, open-source, and Java-based logging library. It provides a robust framework for recording application events, such as errors, warnings, and informational messages. These logs are essential for debugging, monitoring application behavior, and understanding system activity. Log4j offers flexibility by allowing developers to control which log messages are generated, how they are formatted, and where they are sent (for example, to the console, files, or databases).

Log4j 2 is the latest version and offers improvements over its predecessor. It delivers enhanced performance, a plug-in architecture for extensibility, and more flexible configuration options, supporting formats such as XML, JSON, and YAML. Log4j 2 also provides improved support for asynchronous logging, which enhances performance in multithreaded applications.

Here are the key features of Log4j:

- ▶ Asynchronous logging: Improves performance by offloading logging to a separate thread.
- ▶ Plug-in architecture: Enables customization and extension of Log4j's functions.
- ▶ Flexible configuration: Supports various configuration formats (XML, JSON, YAML, and properties).
- ▶ Filters: Allows developers to control which log events are processed.
- ▶ Layouts: Supports various output formats, including custom formats.

To use Log4j in a Java project, complete the following steps:

1. Add Log4j to your project dependencies (through Maven, Gradle, or another product).
2. Create a Logger instance in your Java classes.
3. Log messages by using methods like `logger.debug()`, `logger.info()`, or `logger.error()`.

Example 7-6 provides an example of using Log4j in your Java code.

Example 7-6 Using Log4j

```
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

public class MyApplication {
    private static final Logger logger =
        LogManager.getLogger(MyApplication.class);

    public static void main(String[] args) {
        logger.info("Application started.");
        try {
            // Code logic
            int result = 10 / 0; // Will cause an error
        } catch (Exception e) {
            logger.error("An error occurred: ", e);
        }
    }
}
```

Apache Log4j is an essential tool for managing logging in Java applications. It provides a robust, configurable framework for capturing logs that are used in monitoring, debugging, and error tracking. Keep Log4j updated, especially in response to vulnerabilities such as Log4Shell to maintain security and reliability in production environments.

7.3 Autoscaling and quality of service

A node is considered overcommitted when it has a pod that is scheduled without a resource request, or when the total resource limits of all pods on the node exceed the available machine capacity. In an overcommitted environment, pods might attempt to use more compute resources than are available at any given time. When this situation occurs, the node must prioritize one pod over another one. This decision is based on the quality of service (QoS) class that is assigned to each pod.

7.3.1 Configuring quality of service for pods

You can modify the QoS class in Red Hat OpenShift by changing the resources stanza in the workload's deployment YAML file. In an overcommitted environment, when multiple pods on a node attempt to use more resources than are available, Red Hat OpenShift allocates resources based on the QoS class that is assigned to each pod. This behavior is governed by the CPU and memory allocation algorithm, which determines how physical resources are distributed according to the defined resource settings, as described in [This Red Hat OpenShift document](#).

Red Hat OpenShift established the following three tiers of service:

- ▶ Guaranteed QoS
- ▶ Burstable QoS
- ▶ Best Effort QoS

Table 7-1 summarizes the settings for each tier that is defined in the QoS settings.

Table 7-1 QoS tiers

QoS name	Priority	Resource stanza parameters	Description or Use
Guaranteed	1-highest	limits = requests	High priority, time sensitive tasks.
Burstable	2	limits > requests	Most common workloads. Maximizes instantaneous access to vCPU access.
BestEffort	3-Lowest	Not set	Low-priority tasks. The first to be terminated if the system runs out of resources as part of background system housekeeping.

The QoS setting can impact pod performance. In a recent study², a team observed a substantial improvement after modifying the resources stanza in the workload's deployment YAML file. Specifically, switching from a BestEffort QoS class to a Burstable QoS class enabled Red Hat OpenShift Container Platform to achieve more than twice the throughput for a pod.

² Source:

<https://community.ibm.com/community/user/powerdeveloper/blogs/mithun-h-r/2024/12/13/run-mongodb-and-nodejs-on-red-hat-openshift>

7.4 Management

Managing a modern infrastructure on IBM Power servers requires a strategic approach that incorporates advanced technologies and best practices. One key strategy is cloud integration, which involves migrating workloads to hybrid or multicloud environments. This approach enables organizations to benefit from the scalability, flexibility, and cost efficiency of cloud services while maintaining the robust performance and security of IBM Power servers.

Automation also plays a critical role in streamlining operations, reducing manual intervention, and improving overall efficiency. By automating routine tasks such as system updates, backups, and performance monitoring, IT teams can focus on higher-value initiatives that support business growth.

Another important strategy is data analytics and optimization. IBM Power servers are designed to handle large volumes of data, making them suitable for running complex analytics workloads. By implementing AI-driven analytics tools, organizations can gain deeper insights into operations, optimize resource allocation, and improve decision-making processes. These tools analyze historical data to identify trends, predict future demands, and recommend actions to enhance performance and efficiency.

Security and compliance are also critical aspects of managing modern infrastructure. Although IBM Power servers offer robust built-in security features, integrating AI-driven security solutions enables real-time threat detection and response. This integration helps ensure that systems remain secure and compliant with industry regulations.

The complexities of modern infrastructure require AI-driven tools to meet evolving demands:

- ▶ IBM Watson AIOps is a comprehensive solution that integrates event management, incident diagnosis, incident resolution, and insight delivery into a single platform. It uses AI to predict, communicate, and resolve events before they escalate, improving system reliability and reducing downtime.
- ▶ Red Hat Advanced Cluster Management for Kubernetes provides centralized management of clusters and applications with integrated security policies. It enables organizations to deploy applications, manage multiple clusters, and enforce policies across diverse environments, helping ensure consistency and compliance.

By using AI-driven tools, organizations can manage IBM Power infrastructures more effectively, helping ensure agility, efficiency, and resilience in the face of increasing complexity.

7.4.1 IBM Watson AIOps

IBM Watson AIOps is an AI-powered platform that is designed to automate IT operations (ITOps), predict and prevent outages, and optimize resource usage across complex hybrid cloud environments. It uses AI and ML to analyze large volumes of operational data, identify patterns, and provide actionable insights to IT teams. The primary goal of IBM Watson AIOps is to shift from reactive incident management to proactive problem resolution, improving application availability, performance, and operational efficiency while reducing costs.

At its core, IBM Watson AIOps ingests and correlates data from various IT sources, including monitoring tools (such as IBM Instana and Prometheus), logging systems, event management platforms, service management tools (such as ServiceNow), and change management systems. By applying advanced AI algorithms, it detects anomalies, identifies root causes, predicts potential incidents, and recommends optimal remediation steps.

The platform builds a dynamic, holistic view of the IT environment by understanding the relationships between applications, infrastructure, and services. This contextual awareness is essential for effective problem diagnosis and resolution in today's complex and interconnected IT landscapes.

IBM Watson AIOps combines a set of capabilities into a single solution that helps predict, communicate, and resolve events before they become serious problems. Issues are inevitable in any IT environment, but how and when you respond can make a difference to your organization.

The core functions of IBM Watson AIOps can be categorized into four capabilities: event management, incident diagnosis, incident resolution, and insight delivery. These capabilities are supported by an ecosystem of connectors and tools that manage all aspects of the AIOps lifecycle, from model training to execution.

Figure 7-15 shows how these capabilities map to IBM Watson AIOps.

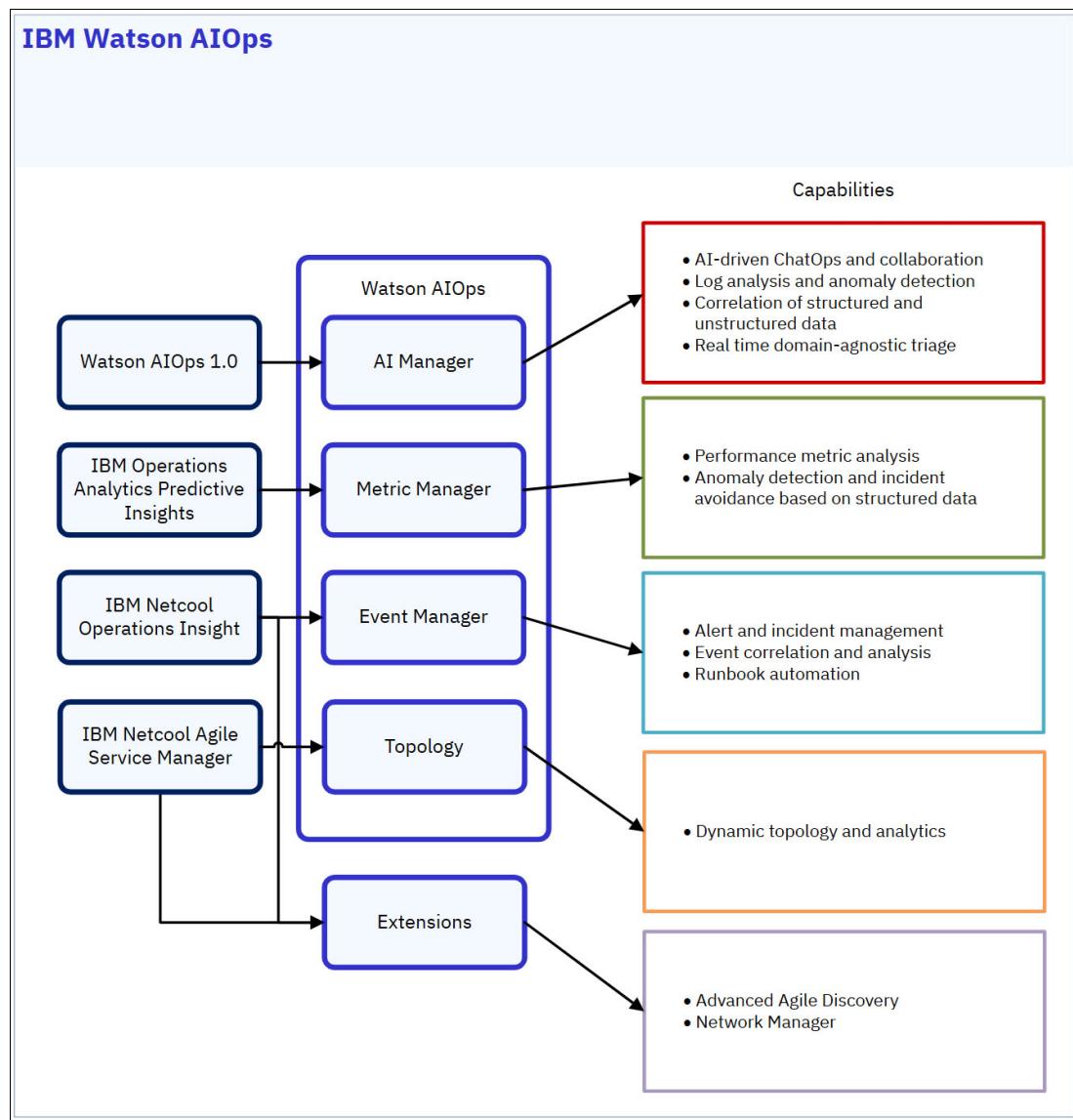


Figure 7-15 Components of Watson AIOps

Note: IBM Watson AIOps is available as an IBM Cloud Pak. For more information, see [IBM Cloud Pak for IBM Watson AIOps](#).

Deriving insight into IT operations with IBM Watson AIOps AI Manager

In today's rapidly evolving technological landscape, Chief Information Officers (CIOs) face an ongoing challenge: fostering innovation while maintaining a stable and reliable IT environment. This balancing act is further complicated by increasing complexity, the need to scale operations efficiently, and the demand for specialized skills to keep pace with constant change. IBM Watsonx AIOps AI Manager addresses these challenges directly, enabling CIOs and Site Reliability Engineers (SREs) to manage complexity with greater agility and confidence.

By using AI, Watsonx AIOps AI Manager intelligently analyzes large volumes of operational data from diverse sources, including system logs, performance metrics, and event streams to uncover hidden patterns and critical insights. This deep analysis goes beyond traditional monitoring approaches by revealing underlying issues and potential risks that might otherwise go unnoticed.

These actionable insights are delivered in near real-time directly into the collaboration and workflow tools that IT teams already use, such as Slack. This integration helps ensure that the right information reaches the right people at the right time, enabling faster awareness and more effective collaboration.

The result is an unprecedented level of visibility into the entire IT infrastructure, providing a holistic understanding of system health and interdependencies. This enhanced visibility enables IT teams to proactively identify potential failures before they affect critical services, shifting from a reactive to a predictive approach.

When issues do occur, Watsonx AIOps AI Manager accelerates problem resolution by delivering intelligent root cause analysis and actionable recommendations. This capability helps minimize downtime, improve application availability, and empower CIOs to pursue innovation on a stable and resilient foundation.

Table 7-2 shows the operational information that is available from IBM Watson AIOps.

Table 7-2 Operational information that is available from IBM Watson AIOps

Operation	Description
Anomaly detection	Detects anomalies from data (real-time or offline).
Event grouping	Groups related events to aid incident diagnosis. Events can include, for example, Pager Duty alerts, IBM Netcool® Operations Insight® alerts, or log anomalies.
Blast radius and fault localization	Derives the root fault component, and derives the full scope of components that are affected by an incident.
Incident similarity	For a particular incident, finds the highest n -ranked similar incidents from the past.
Next best action	For a particular incident, suggests the highest n actions from similar incidents from the past.

7.4.2 Red Hat Advanced Cluster Manager

Administrators and SREs often face challenges when working across diverse environments, including multiple data centers, private clouds, and public clouds running Kubernetes clusters. Red Hat Advanced Cluster Management for Kubernetes provides the tools and capabilities that are needed to address these common challenges.

Red Hat Advanced Cluster Management for Kubernetes provides end-to-end management visibility and control to manage your Kubernetes environment. Take control of your application modernization program with management capabilities for cluster creation, application lifecycle, and provide security and compliance for all of them across data centers and hybrid cloud environments.

Red Hat Advanced Cluster Management for Kubernetes provides end-to-end visibility and control for managing Kubernetes environments. It supports application modernization by enabling cluster creation, managing application lifecycles, and enforcing security and compliance policies across data centers and hybrid cloud environments.

Red Hat Advanced Cluster Management for Kubernetes provides centralized, end-to-end visibility and control for managing Kubernetes environments. Clusters and applications are visible and managed from a single console with integrated security policies. You can operate from any location where Red Hat OpenShift runs and manage any Kubernetes cluster. It offers the following capabilities:

- ▶ Provides end-to-end management, visibility, and control of the cluster and application lifecycle. Enhances security and compliance across the entire Kubernetes domain, including multiple data centers and public cloud environments.
- ▶ Provides a hybrid cloud management platform with capabilities that address common challenges that faced by administrators and SREs. Supports operations across diverse environments, including multiple data centers, private and public clouds, and remote edge sites running Kubernetes clusters.
- ▶ Provides FIPS mode support and enables centralized management of Kubernetes clusters. These clusters can span public clouds (AWS, Google Cloud, or Microsoft Azure), private clouds (OpenStack, or virtualization platforms), on-premises environments (bare metal servers with x86_64, IBM Power, and IBM Z and LinuxONE systems), and edge environments (ARM-based systems).
- ▶ Delivers multi-cluster observability for monitoring cluster health and optimizing performance. Provides dashboards that store long-term historical data and offer a comprehensive view of your entire cloud-ready environment.
- ▶ Enables unified multi-cluster lifecycle management to create, upgrade, and decommission Kubernetes clusters reliably, consistently, and at scale. Uses an open-source programming model that supports and promotes infrastructure as code (IaC) best practices and design principles.
- ▶ Enables policy-based governance, risk, and compliance by automatically monitoring and enforcing best-practice configurations. Helps ensure that controls that are related to security, resiliency, and software engineering align with industry compliance standards or internal corporate policies.
- ▶ Supports advanced application lifecycle management by integrating open standards and deploying applications through placement rules. These rules are embedded into existing continuous integration and continuous deployment (CI/CD) pipelines and governance controls.

- ▶ Enables edge management at scale by using single-node Red Hat OpenShift clusters and Red Hat Advanced Cluster Management. Supports continuous scaling and high availability (HA) in edge environments with high latency and low bandwidth.
- ▶ Supports business continuity by using Red Hat Advanced Cluster Management and the broader Red Hat portfolio to help ensure that critical applications, including stateful workloads, remain continuously available.

Architecture

Red Hat Advanced Cluster Management for Kubernetes includes several multi-cluster components that are used to access and manage clusters. The term *hub cluster* refers to the central controller that runs within a Red Hat Advanced Cluster Management for Kubernetes environment. From the hub cluster, you can access the console, product components, and APIs such as the rcm-api, which handles requests that relate to cluster lifecycle management.

Figure 7-16 illustrates the hub cluster and a single managed cluster. Red Hat Advanced Cluster Management enables centralized management of multiple clusters from a single hub cluster.

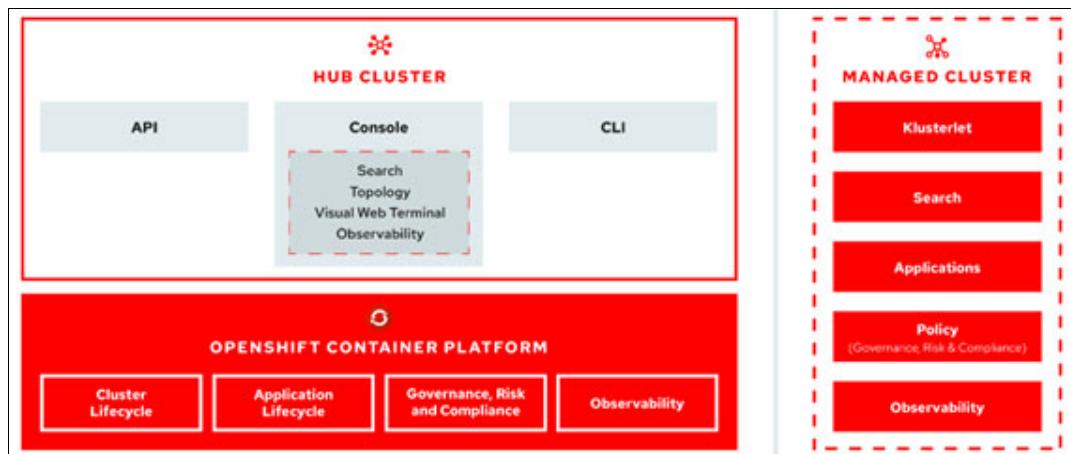


Figure 7-16 Red Hat Advanced Cluster Management hub cluster and managed cluster³

The hub cluster aggregates information from multiple clusters and maintains the state of managed clusters and their applications. Red Hat Advanced Cluster Management provides a set of REST APIs to support its management functions. A managed cluster is any cluster that is controlled by the hub cluster. The Klusterlet agent manages the connection between the managed cluster and the hub cluster.

Figure 7-17 on page 231⁴ shows a more detailed view of the Red Hat Advanced Cluster Management components.

³ Source:

https://docs.redhat.com/en/documentation/red_hat_advanced_cluster_management_for_kubernetes/2.1/html/about/welcome-to-red-hat-advanced-cluster-management-for-kubernetes#welcome-to-red-hat-advanced-cluster-management-for-kubernetes

⁴ Source:

https://docs.redhat.com/en/documentation/red_hat_advanced_cluster_management_for_kubernetes/2.2/html/security/governance-and-risk

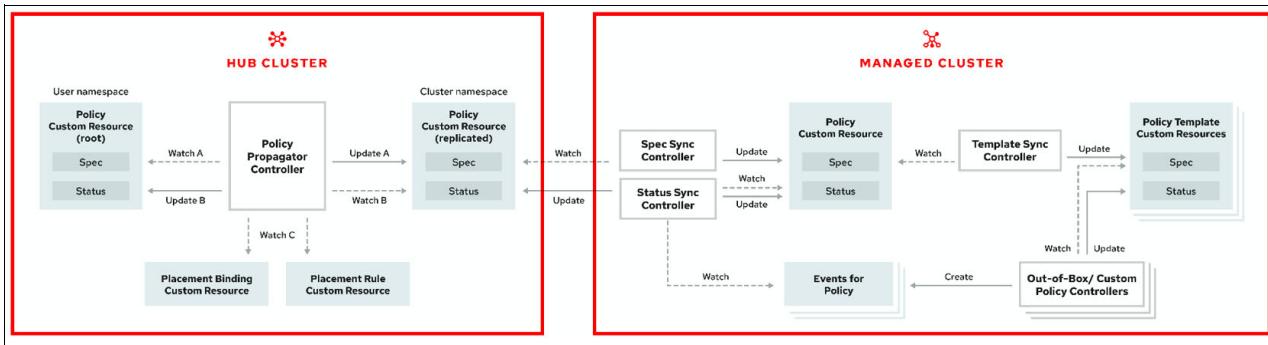


Figure 7-17 Cluster components

The Red Hat Advanced Cluster Management hub cluster acts as a central controller, providing access to the console and product components. It uses APIs to manage cluster lifecycle operations, including resource discovery across clusters, command execution through the Visual Web Terminal, and topology visualization. The hub cluster also enables observability by collecting metrics from managed clusters and cloud providers.

The hub cluster aggregates information from multiple clusters by using an asynchronous work request model. It maintains the state of clusters and the applications running on them by using a graph database. It also uses etcd, a distributed key-value store to persist the state of work requests and results from multiple clusters.

The managed cluster receives and processes requests, then returns the results to the hub cluster. It also sends metrics to the hub cluster through the observability service.

Red Hat Advanced Cluster Management can create and import clusters, and manage cluster upgrades and decommissioning across on-premises, public, and private cloud environments. It provides an aggregated view of overall cluster health, and detailed metrics for individual managed clusters.

Red Hat Advanced Cluster Management supports application lifecycle management and can manage application resources across managed clusters. It automates the deployment and lifecycle operations of these resources. It also exposes reconcilable service routes and provides full control over Kubernetes resource updates to manage all aspects of the application.

Red Hat Advanced Cluster Management enables centralized governance and risk management by defining processes to manage security and compliance from a single interface. Cluster administrators can view and create policies by using the Red Hat Advanced Cluster Management policy framework, configure role-based access control (RBAC), and define security policies.

The observability component helps monitor the health and usage of clusters across a fleet. By default, the multicluster observability operator is enabled during the installation of Red Hat Advanced Cluster Management. Observability includes several key components:

- ▶ Observability architecture
- ▶ Observability configuration
- ▶ Enabling the observability service
- ▶ Using observability
- ▶ Customizing observability
- ▶ Observability alerts
- ▶ Console search introduction
- ▶ Using observability with Red Hat Insights

Observability architecture

You can use Red Hat Advanced Cluster Management for Kubernetes to gain insights and optimize managed clusters. To monitor cluster health, enable a `multicloud-observability-operator`, which provides observability services. Although observability is included with the product installation by default, it is not enabled automatically due to the requirement for persistent storage.

The `multicloudhub-operator` enables the `multicloud-observability-operator` pod by default. Users must configure `multicloud-observability-operator` to activate observability. When the Observability service is enabled by defining a `MultiCloudObservability` custom resource, the `observability-endpoint-operator` is automatically deployed to each imported or created managed cluster.

This controller starts a metrics collector that gathers data from Red Hat OpenShift Container Platform Prometheus and sends it to the Red Hat Advanced Cluster Management hub cluster.

Figure 7-18 illustrates the observability components.

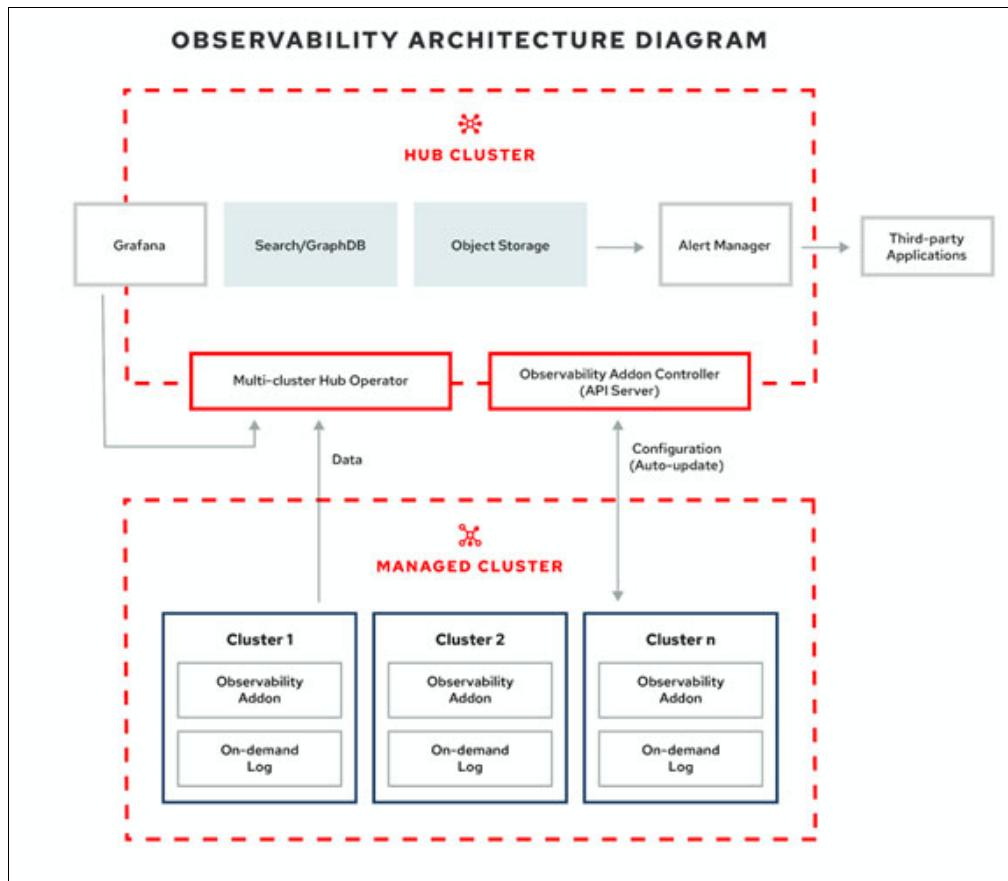


Figure 7-18 Observability architecture diagram⁵

⁵ Source:

https://docs.redhat.com/en/documentation/red_hat_advanced_cluster_management_for_kubernetes/2.3/html-single/observability/index#observing-environments-intro

When the Observability service is enabled, the hub cluster is always configured to collect and send metrics to the designated Thanos instance, regardless of whether hub self-management is enabled. If the hub cluster is self-managed, the `disableHubSelfManagement` parameter is set to `false`, which is the default. Metrics and alerts for the hub cluster appear in the local-cluster namespace. The local-cluster appears in the cluster list drop-down menu only when hub self-management is enabled. You can query local-cluster metrics by using the Grafana explorer.

The components of the observability architecture include the following items:

- ▶ The multicluster hub operator, also known as the `multiclusterhub-operator` pod, deploys the `multicluster-observability-operator` pod. It sends hub cluster data to managed clusters.
- ▶ The observability add-on controller acts as the `apiserver` and automatically updates the log of the managed cluster.
- ▶ The Thanos infrastructure includes the Thanos Compactor, which is deployed by the `multicluster-observability-operator` pod. The Thanos Compactor improves query performance by applying retention policies and compacting stored data. To help identify potential issues, four default alerts monitor the health of the Thanos Compactor.

The observability component deploys an instance of Grafana to support data visualization through static dashboards and data exploration. Red Hat Advanced Cluster Management supports Grafana 8.5.20. You can also design custom Grafana dashboards. For more information, see [Designing your Grafana dashboard](#).

Alertmanager

Prometheus Alertmanager enables integration with third-party applications by forwarding alerts. You can customize the Observability service by creating custom recording or alerting rules. Red Hat Advanced Cluster Management supports Prometheus Alertmanager 0.25.

Figure 7-19 illustrates how the Alertmanager functions.

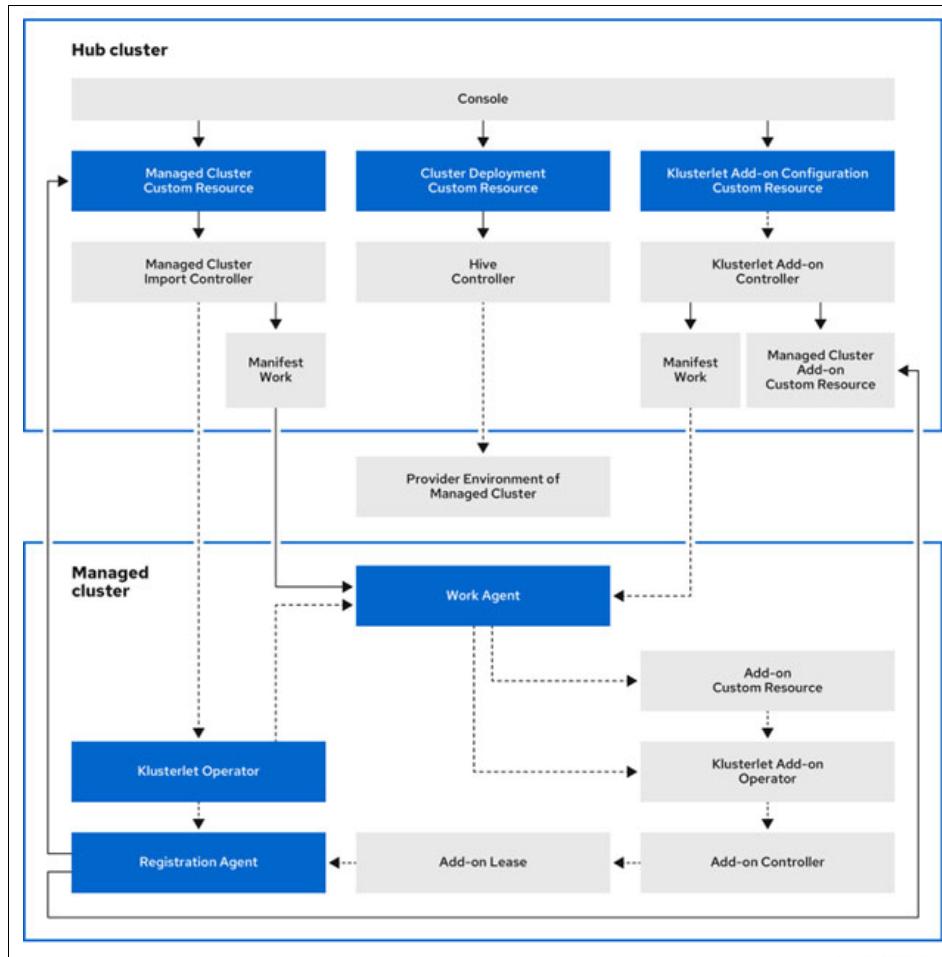


Figure 7-19 High-level Alertmanager⁶

There are predefined alerts that are managed. For a list of default alerts, see Table 7-3.

Table 7-3 List of default alerts:

Alert	Severity	Description
ACMThanosCompactHalted	Critical	An alert is sent when the compactor stops.
ACMThanosCompactHighCompactionFailures	Warning	An alert is sent when the compaction failure rate is greater than 5%.
ACMThanosCompactBucketHighOperationFailures	Warning	An alert is sent when the bucket operation failure rate is greater than 5%.
ACMThanosCompactHasNotRun	Warning	An alert is sent when the compactor has not uploaded anything in the last 24 hours.

⁶ Source:

https://docs.redhat.com/en/documentation/red_hat_advanced_cluster_management_for_kubernetes/2.5/html/clusters/managing-your-clusters

Sizing for Red Hat Advanced Cluster Management

Table 7-4 shows the sizing requirements for Red Hat Advanced Cluster Manager. These sizes are the suggested minimum compute resources in addition to any Red Hat OpenShift Container Platform requirements.

Table 7-4 Minimum sizes of compute resources

Node role	Minimum no. of nodes	Data stores	Total reserved memory (lower bound) per node	Total reserved CPU (lower bound) per node
Master	3	etcd x 3	Per the Red Hat OpenShift Container Platform sizing guidelines	Per the Red Hat OpenShift sizing guidelines
Worker	3	redisgraph redis x 1	16 GB	6 CPUs

Modernization support by operating system

Modernization occurs across multiple layers of the IBM Power infrastructure. IBM has integrated modern tools and capabilities into the Power hardware, enhanced the management layer, and introduced technologies and frameworks to support application modernization. This section examines improvements to the operating systems (OSs) running on IBM Power, specifically AIX, IBM i, and Linux. These OSs continue to evolve by incorporating advancements in security, encryption, automation, and support for modern application development tools. These enhancements are designed to help clients migrate to a modernized infrastructure.

The following chapters are included in this part:

- ▶ Chapter 8, “AIX” on page 239
- ▶ Chapter 9, “IBM i” on page 279
- ▶ Chapter 10, “Linux and Red Hat OpenShift” on page 343



AIX

IBM AIX is a robust and reliable operating system (OS) for mission-critical applications in enterprise environments. With more than 30 years of performance, AIX offers a stable, secure, and scalable platform that supports businesses across diverse industries.

AIX on IBM Power servers supports innovation through hybrid cloud and open-source capabilities, providing businesses with a secure and resilient environment for building and deploying modern applications. As the foundation for many core business applications and database systems, AIX continues to evolve by integrating new hybrid multicloud features and enhanced open-source support.

Customers that use IBM AIX benefit from improved workload scalability, enhanced cloud automation with Ansible, strengthened security, flexible licensing options, and access to more than 300 open-source packages. IBM Power servers remain committed to delivering ongoing enhancements to AIX, with a focus on performance, scalability, resilience, and continuous innovation.

The following topics are described in this chapter:

- ▶ 8.1, “Introducing AIX” on page 240
- ▶ 8.2, “AIX advantages” on page 240
- ▶ 8.3, “AIX 7.3” on page 245
- ▶ 8.4, “AIX processor compatibility modes” on page 248
- ▶ 8.5, “AIX and Power10 support” on page 250
- ▶ 8.6, “AIX support for open source” on page 252
- ▶ 8.7, “Live Update” on page 263

8.1 Introducing AIX

IBM AIX is a proprietary UNIX like OS that is developed by IBM for IBM Power servers. Introduced in 1986 and based on UNIX System V, AIX continually evolves. The latest version, AIX 7.3, provides a stable and reliable platform for mission-critical workloads and has supported enterprise computing for more than 30 years.

IBM Power remains at the forefront of innovation, prioritizing performance, resilience, scalability, and security to maintain its position as a leading server platform. AIX protects customer investments through binary compatibility and long release lifecycles. It also supports the adoption of modern technologies with flexible, subscription-based models.

In response to increasingly sophisticated cyberthreats, AIX 7.3 includes advanced security enhancements to protect data. IBM PowerSC further strengthens security by addressing complex threats and misconfigurations, simplifying administrative tasks, and streamlining compliance efforts.

As IBM Power expands its capabilities in hybrid cloud, artificial intelligence (AI), and cloud-native applications, AIX remains a critical part of the strategy. With a roadmap and support plan extending beyond 2035, IBM demonstrates its commitment to the long-term viability of AIX. The current roadmap is shown in Figure 8-1.

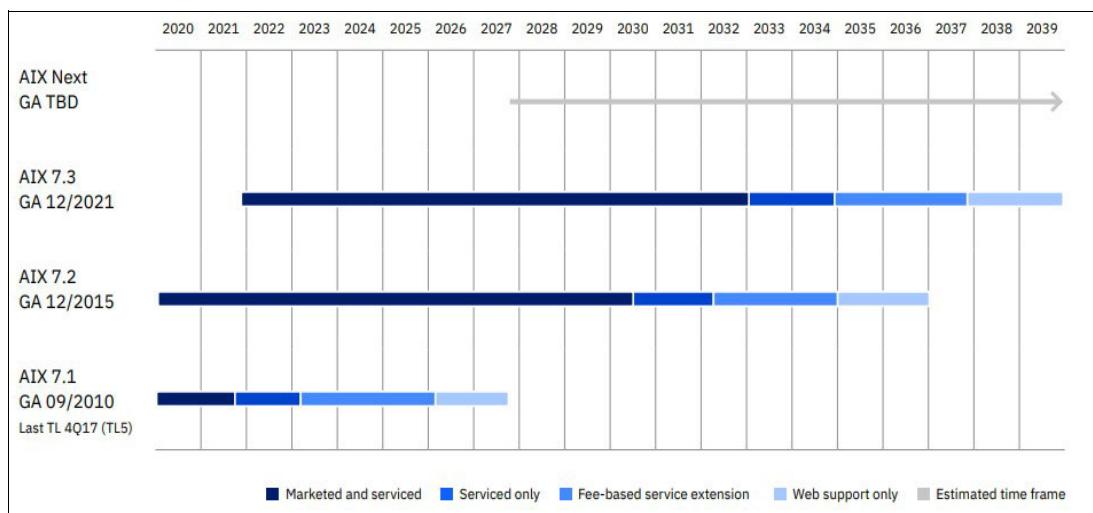


Figure 8-1 AIX roadmap

For more information about IBM's strategy and roadmap for IBM AIX, see [An executive guide to the strategy and roadmap for the IBM AIX operating system for IBM Power servers](#).

8.2 AIX advantages

IBM is committed to supporting the thousands of businesses that rely on AIX for their most mission-critical applications and databases. This commitment has positioned AIX as a market leader in scalable, distributed operating environments across key industries, such as banking, insurance, telecommunications, retail distribution, health care, and the public sector. AIX is trusted for its consistent delivery of high performance, scalability, availability, and security for essential workloads.

With a roadmap and ongoing support extending beyond a decade, IBM reinforces its commitment to innovation, particularly in hybrid cloud, AI, and other emerging technologies, helping ensure that AIX continues to meet the evolving needs of businesses and their users.

Figure 8-2 summarizes the AIX core strengths and strategic directions.

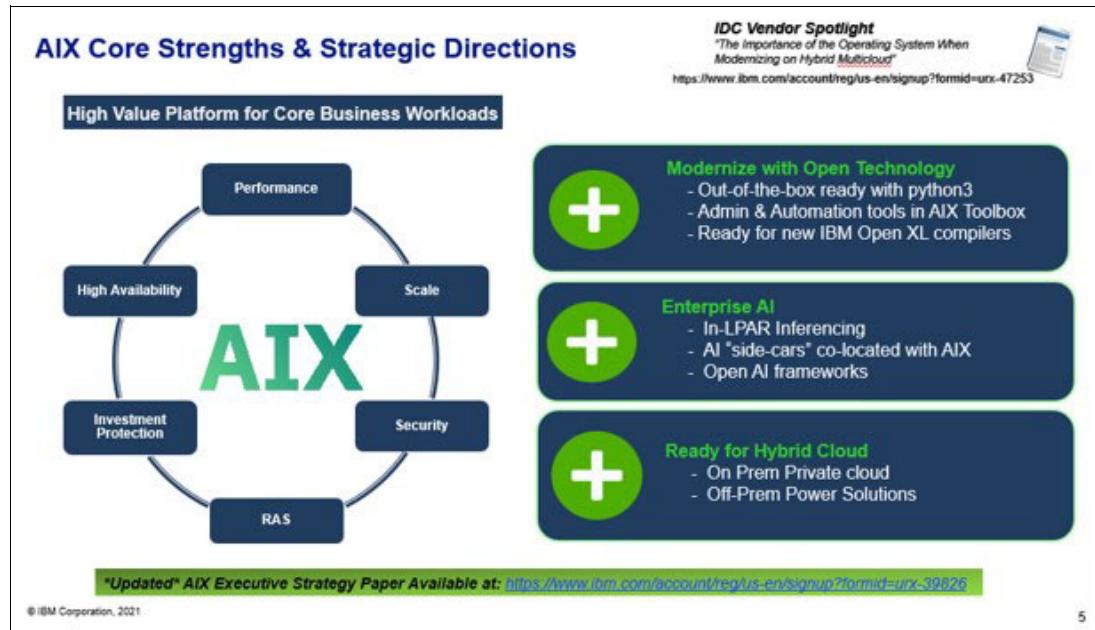


Figure 8-2 AIX Core Strengths & Strategic Directions

8.2.1 Reliability, availability, and serviceability

In computing, reliability, availability, and serviceability (RAS) are essential attributes of a computer system, particularly in enterprise and mission-critical environments. Here is a breakdown of each component:

- **Reliability:** The system's ability to perform its intended functions without failure over a specified period. A reliable system minimizes errors and hardware failures, helping ensure consistent performance.
- **Availability:** The system's ability to remain operational, accessible, and ready for use when needed. High availability (HA) helps ensure that services and applications remain available to users with minimal downtime, often through redundancy, fault tolerance, or failover mechanisms.
- **Serviceability:** The ease with which a system can be maintained and repaired. Serviceable systems are designed for efficient diagnostics, repairs, and upgrades, minimizing downtime and extending hardware lifespan.

RAS is a critical set of features for systems that are used in industries where downtime is costly or disruptive, such as data centers, telecommunications, and enterprise environments. Systems with strong RAS characteristics aim to deliver uninterrupted service, recover quickly from failures, and maintain ongoing operations with minimal administrative effort.

For AIX running on IBM Power, RAS is a core design priority. These servers are built to deliver the high standards of performance, uptime, and maintainability that are required by enterprise and mission-critical workloads. IBM Power servers, which are used in industries such as banking, telecommunications, and health care, incorporate robust RAS features to maximize system efficiency and minimize downtime.

Reliability

Reliability is the consistency with which a system performs its intended tasks without failure. AIX running on IBM Power servers integrates hardware and software features that work together to deliver a high level of reliability.

Here are some key elements that contribute to reliability:

- ▶ Advanced hardware design

IBM Power servers include advanced error detection and correction capabilities at the hardware level. Features such as memory mirroring and error-correcting code (ECC) memory help prevent data corruption by automatically detecting and correcting memory errors.

- ▶ Fault-tolerant components

IBM Power servers are engineered with redundant components, such as power supplies and cooling units to eliminate single points of failure. In addition, Power10 processors include capabilities to detect and mitigate errors at the chip level, helping reduce application downtime. These features enable the system to continue operating smoothly even if individual components fail.

- ▶ AIX reliability

The AIX OS delivers software reliability through infrequent failures and robust recovery capabilities, including self-healing mechanisms. When an issue occurs, AIX does not stop running, but instead attempts to continue operating, self-heals the affected process, and generates alerts and logs that describe the condition. Built-in features such as the System Resource Controller, Logical Volume Manager (LVM), Error Reporting, and Reliable Scalable Cluster Technology (RSCT) help AIX achieve this level of resilience.

Availability

Availability focuses on ensuring that the system is continuously operational and accessible when needed. IBM Power servers are designed for HA, which is crucial in industries requiring continuous operations.

IBM incorporates several features to maximize availability:

- ▶ Live Partition Mobility (LPM)

AIX supports LPM, which enables workloads to move between servers without downtime. This capability facilitates load-balancing, proactive maintenance, and disaster recovery (DR).

- ▶ PowerHA

IBM PowerHA provides high availability and disaster recovery (HADR) solutions for AIX environments. It helps ensure that applications and data remain accessible during hardware or software failures by enabling clustering and automated failover.

- ▶ Automatic restart and self-healing

IBM Power servers include automatic restart features that detect faults and restart components to restore functions. The system can self-diagnose and resolve many issues without human intervention, helping ensure continuous availability.

- ▶ Live Kernel Update (LNU)

AIX features a dynamic kernel that supports runtime modifications without requiring compilation or restart. The kernel is extendable and can be expanded by adding routines from any functional class. A process running in user mode can customize the kernel by using the sysconfig subroutine if it has the necessary privileges. This situation enables the process to load, unload, initialize, or terminate kernel routines. Kernel configuration can also be modified by adjusting tunable system parameters.

These kernel extensions can include additional kernel services, system calls, device drivers, or file systems that are related to OS and device management. The sysconfig subroutine also enables reading and setting system runtime operating parameters.

Note: For more information about the abilities of the AIX kernel, see the following resources:

- ▶ [Kernel Environment](#)
- ▶ [The sysconfig Subroutine](#)
- ▶ [Kernel Extensions and Device Support Programming Concepts](#)

Serviceability

Serviceability is about how quickly a system can be repaired, maintained, and upgraded, minimizing downtime for administrators and helping ensure the system's longevity. IBM Power servers are designed with serviceability in mind through various features:

- ▶ Predictive Failure Analysis (PFA)

IBM Power servers include PFA, which detects and reports potential issues before they lead to failures. This capability enables administrators to proactively replace components, reducing unplanned downtime and preventing service interruptions.

- ▶ First Failure Data Capture (FFDC)

FFDC is a mechanism that captures critical system information when a severe, unrecoverable error occurs. Its primary purpose is to provide IBM Support teams with the data that is needed to diagnose and resolve critical issues quickly and effectively. When AIX encounters a serious fault that prevents normal operation, FFDC logs essential system details to aid in troubleshooting and root cause analysis.

- ▶ Remote diagnostics and management

IBM Power servers include a suite of remote management tools, such as IBM Systems Director and the Integrated Management Module (IMM), which enable administrators to monitor system health, identify issues, and manage hardware and software remotely. These tools help improve system uptime and reduce the time that is required for onsite service.

- ▶ Hot-swappable components

Many IBM Power servers enable you to replace components such as power supplies, fans, and hard disk drives without shutting down the system. This capability enables faster repairs and upgrades without affecting system availability.

8.2.2 Scalability and dynamic management

AIX excels in dynamic adaptability. It integrates new devices and accommodates configuration changes in real time, minimizing disruptions and maintaining optimal performance. To enhance uptime and scalability, AIX provides robust HA clustering, enabling you to distribute workloads and help ensure continuous operation.

RSCT/RMC

The RSCT/RMC implementation provides a HA infrastructure for managing resources in both stand-alone systems and clusters (peer domains or management domains). RSCT is a set of software components that together deliver a comprehensive clustering environment for AIX and Linux. Various IBM products use RSCT to support cluster infrastructures that improve system availability, scalability, and usability.

RSCT includes the following components:

- ▶ The Resource Monitoring and Control (ctrmc) subsystem
- ▶ RSCT core resource managers (ctcas)
- ▶ SCT cluster security services (ctsec)
- ▶ The Topology Services subsystem (cthatsd)
- ▶ The Group Services subsystem (cthagsd)

Note: For more information, see [RSCT concepts](#).

Object Data Manager

The ODM is a unique feature of the AIX OS. It is a data manager that is designed to store system information as objects with associated characteristics. This mechanism helps ensure the integrity of AIX by preventing administrators from configuring unsupported device or system attributes. ODM is customizable and contributes to the extensibility of AIX.

Other UNIX like systems use ASCII stanza files that administrators manually edit to configure drivers in the kernel. In contrast, AIX uses stanza files but stores them in a directory that is known as the ODM database. Unlike other systems, you do not directly modify these files in AIX. Instead, you use ODM routines and commands to make changes. Typically, you do not need to run ODM commands manually because standard system utilities such as `cfgmgr`, `mkdev`, and `rmdev` automatically start the ODM routines to update the database.

A device configuration method in AIX is started either by a root user running the `cfgmgr` command or by the `rc.boot` script in the RAM disk during system startup. This method eventually calls the sysconfig subroutine that was referenced earlier.

The following list provides the device methods that are called during a `cfgmgr` task and the interaction with the AIX ODM when a device is added to the AIX OS:

- ▶ Define method (causes device to be defined)
A define method's main task is to retrieve device data from PdDv in ODM and create a CuDv object. Also, it helps ensure that a parent device exists in the CuDv object.
- ▶ Configure method (causes device to be available)
A configure method should perform the following steps:
 - a. Display an LED value on the system LED panel.
 - b. Verify that a parent device is available (in ODM).
 - c. Verify that a device is present.

- d. Start the **busresolve** system call to get an interrupt level assigned to the device.
- e. Extend the kernel by calling sysconfig.

Note: For more information about the operation of device drivers in AIX, see this [PDF on writing device drivers](#).

This IBM document provides an example showing how an update in the AIX ODM manages the changes to the AIX kernel to enable dynamic tracking of Fibre Channel (FC) devices.

For more information about ODM concepts, see the following resources:

- ▶ [AIX](#)
- ▶ [AIX Version 7.2: Customizing AIX ODM for PCI/PCIe adapters](#)

8.2.3 Advanced virtualization

AIX is designed as the flagship OS for IBM Power, where the advanced PowerVM hypervisor supports mission-critical business requirements.

IBM PowerVM is a robust virtualization technology that enables IBM Power servers to run multiple virtual machines (VMs) concurrently. This capability enables you to consolidate workloads, improve server usage, and reduce costs. PowerVM provides a secure and scalable virtualization environment that supports a range of operating systems, including AIX, IBM i, and Linux.

PowerVM creates logical partitions (LPARs) within a physical server, with each partition functioning as an independent system. This structure enables flexible resource allocation, enabling processors, memory, and I/O to be dynamically assigned to individual VMs based on specific requirements. PowerVM also provides advanced features such as LPM, which enables you to migrate running VMs between servers without downtime, enhancing both availability and flexibility. For more information, see *Introduction to IBM PowerVM*, SG24-8535.

8.3 AIX 7.3

AIX 7.3 introduces a suite of enhancements that are designed to meet the evolving demands of modern IT environments. Performance is a key focus, with optimizations that use the latest IBM POWER processor capabilities to improve system speed, memory management, and I/O operations. Security is also strengthened, with features that protect against emerging threats and extend protection to virtualized environments. Reliability and availability are enhanced through improvements to HA clustering and overall system stability. Management is streamlined with improved tools and automation support, simplifying administrative tasks. AIX 7.3 also embraces modern technologies by offering enhanced support for containerization and cloud environments, along with updates to core file system functions and size limitations. A notable enhancement is the refinement of the Live Update feature, which enables more dynamic and less disruptive changes to running LPARs.

Here is a summary of some of the enhancements that are delivered in AIX 7.3:

- ▶ Extends the scalability of AIX:
 - Supports a maximum of 240 cores.
 - Increased file and file system capacity to 128 TB for growing data needs.
- ▶ Reduced downtime:
 - Reduced initial program load (IPL) times for large memory LPARs.
 - vPMEM enables the persistence of memory during a VM restart, and avoids time-consuming memory reloads for in-memory applications like SAP HANA.
 - LKU improvements:
 - A reduced “blackout” window
 - Overall performance improvements
 - Enable IPsec and audit stream for LKU
 - Live LPAR profile updates
 - Console message improvements
 - Live Library Update (LLU).
 - Dump time improvements.
- ▶ Workload optimization:
 - Integrates the usage of an on-chip NZ GZIP accelerator with AIX16 MB MPSS for text segments.
 - Virtual Ethernet performance.
 - Direct-attach I/O performance.
 - Enhanced performance for dynamic compute and memory management (dynamic logical partitioning (DLPAR)).
- ▶ Optimized security:
 - A more secure password policy and algorithm.
 - OpenSSL performance.
 - Audit subsystem performance.
 - IPsec performance.
 - Extends logical volume (LV) encryption to other devices, including rootvg.
 - LV and persistent volume (PV) encryption performance.
 - Opencryptoki.
 - OpenSSH 9.7p1.
- ▶ Administrative efficiency:
 - New automation use cases.
 - AIX Toolbox for Open Source Software updates.
 - Ready for Ansible with Python 3.
- ▶ Streamlines insights and automation:
 - AI inferencing at the point of data with in-core MMA.
 - ESSL 7.1 support with AIX 7.3 and OpenXLC/C++ compilers.
 - AI inferencing with Python and other open-source packages.

For more information about what is new in AIX 7.3, see this [IBM Documentation article](#).

8.3.1 Security updates

AIX 7.3 introduces key security enhancements that strengthen system integrity and protect against emerging threats. One major improvement is the implementation of more robust controls for managing user access. These controls include support for advanced authentication mechanisms, such as two-factor authentication (2FA), and tighter integration with IBM Security Identity Governance and Intelligence solutions. AIX 7.3 also improves the security of system communications by enhancing encryption capabilities, including expanded support for Transport Layer Security (TLS) protocols. These enhancements help ensure that data in transit is more securely protected.

AIX 7.3 enhances the Trusted AIX feature to help prevent unauthorized changes to system configurations. The OS includes advanced auditing capabilities that enable administrators to track system activities with greater granularity. New security policies, such as stricter user account management and improved control over administrative privileges, reduce the attack surface and increase resistance to unauthorized access. These features, along with ongoing updates and fixes from IBM, reinforce AIX 7.3 as a secure platform for enterprise computing.

Here are some of the IBM AIX 7.3 security enhancements:

- ▶ New default password algorithm (SSHA-256).
- ▶ Long password support (up to 255 characters).
- ▶ Stronger default password policy.
- ▶ Revised bos.net installation defaults to omit packages without stronger security.
- ▶ LVM encryption for rootvg and dump devices.
- ▶ RFC 7383 IKE fragmentation with IPsec.
- ▶ IPsec support for Network Address Translation-Traversal (NAT-T) with Internet Key Exchange version 2 (IKEv2).
- ▶ Tracing channel 0 now requires root privilege by default.

For more information about new features in AIX 7.3, see [What's new in AIX security](#).

Trusted Execution

The removal of Trusted Computing Base (TCB) in AIX 7.2 and 7.3 marks a shift toward a more adaptable security model. Although TCB provided a foundation for system integrity, it was a static feature that was enabled during installation, and it offered limited flexibility for evolving security requirements. AIX now emphasizes Trusted Execution (TE), a dynamic set of features that verify system integrity and enforce advanced security policies. TE provides a more granular and flexible approach, enabling administrators to tailor security measures to specific needs. This transition supports a security model that adapts to emerging threats and evolving compliance standards, enhancing the overall trustworthiness of the AIX environment.

A common method that is used to compromise a system involves gaining unauthorized access, installing Trojans or rootkits, and tampering with critical security files. These actions make the system vulnerable to exploitation. The primary goal of TE features is to prevent such malicious activities. In worst-case scenarios, TE features help identify whether an incident occurred, helping ensure that the system remains secure and that any breaches are detectable.

TE is a built-in feature of AIX that verifies the integrity of system executable files and kernel extensions that are permitted to run or load. Similar to TCB, TE uses a database to store critical security parameters for trusted files on the system. This database, which is called the Trusted Signature Database (TSD), is in /etc/security/tsd/tsd.dat.

TE also supports auditing the system's security state and identifying modified files. These capabilities increase the system's trust level and make it more difficult for malicious users to compromise the environment.

The set of features under TE can be grouped into the following categories:

- ▶ Managing TSD
- ▶ Auditing the integrity of the TSD
- ▶ Configuring security policies
- ▶ Trusted Execution Path and Trusted Library Path

Enabling Trusted Execution

For more information about enabling TE, see this [IBM Support document](#). To centrally manage the TE feature of multiple AIX endpoints, use IBM PowerSC. For more information about PowerSC and TE, see *Simplify Management of IT Security and Compliance with IBM PowerSC in Cloud and Virtualized Environments*, SG24-8082.

AIX Security Assessment

Organizations that use AIX or Virtual I/O Server (VIOS) often face challenges in securing their systems. The AIX Security Assessment provides a comprehensive analysis of a single AIX or VIOS instance and requires only a few hours to complete. This offering is designed to identify security safeguards that can be implemented to reduce risk and strengthen the security posture of AIX or VIOS environments.

The assessment consists of two components:

- ▶ Policy Assessment (optional)
- ▶ Host Assessment

For more information about requesting an AIX Security Assessment from IBM Services, see [AIX Security Assessment](#).

8.4 AIX processor compatibility modes

IBM POWER processors support compatibility modes that help ensure smooth migrations and continued operation of applications. These modes allow newer Power servers to run code that was compiled for earlier architectures, minimizing the need for extensive application recompilation or migration. This compatibility with earlier versions is essential for organizations that invested in software that is tailored to specific POWER processor generations.

POWER processors can switch between different instruction set architectures (ISAs), effectively emulating the behavior of earlier generations. This capability enables the running of binary files that were compiled for older Power architectures on the latest hardware. It provides a bridge for customers, enabling them to upgrade their infrastructure while preserving the functions of existing software investments.

Figure 8-3 shows the AIX versions and capabilities that are provided for processor mode on IBM Power8®, IBM Power9, and IBM Power10.

	AIX 7.1 TL5		AIX 7.2 TL5		AIX 7.3 TL0		
	P8 mode	P8 mode	P9 mode	P8 mode	P9 mode	P10 mode	
Max HW threads per lpar	1024	1536	1536	1920	1920	1920	
Max RAM per lpar (TB)	16	32	32	32	32	32	
SMT default	8	8	8	8	8	8	
HW GZIP enabled	No	No	Yes	No	Yes	Yes	
Power10 MMA support	No	No	No	No	No	Yes	
P10 optimized <u>memcpy</u>	No	No	No	No	No	Yes	
XIVE support	No	No	Yes	No	Yes	Yes	

- Realize the benefits of Power10 with the flexibility to choose your AIX level and processor mode
- Move to AIX 7.3 for new capabilities in workload scale, infrastructure optimization, security, and accelerated AI

Figure 8-3 AIX version choices for functions

The Power10 processor is available in both single-chip module (SCM) and dual-chip module (DCM) configurations.

The IBM Power E1080 SCM configuration supports up to 15 cores per socket and a maximum of 16 sockets, providing a total of 240 cores. Each node contains four sockets, and a system can include up to four nodes.

The IBM Power E1050 DCM configuration supports up to 96 cores by using 24-core DCMs across four sockets. Each DCM occupies one socket and is available with 12, 18, or 24 cores. The system supports a maximum of one node with four sockets, and multi-node configurations are not supported.

The hardware design defines near, far, and distant NUMA memory and CPU operations. As access moves from near to far and distant hardware locations, the CPU resources that are required increase.

The AIX OS on IBM Power servers is optimized for general performance through extensive testing and default tuning by the IBM Power Performance team. These defaults are validated against industry-standard benchmarks, including SPEC, TPC, high-performance computing (HPC), and IBM Power performance report (rPerf) data. As a result, AIX can deliver optimal performance without requiring customization.

To prevent unexpected results when administrators modify system settings, the Power Performance team introduced restricted tunables in AIX to improve the code base. These tunables can be changed only after a detailed workload analysis and approval from AIX Performance Development. This process might result in a code update or a modification to the default tunable value.

The AIX version functions are automatically implemented by PowerVM by the processor compatibility mode that is determined by the AIX OS version that is installed.

The hypervisor sets the effective processor compatibility mode for an LPAR by using the following information:

- ▶ The processor features that are supported by the OS environment running in the LPAR.
- ▶ The configured processor compatibility mode that you specify.

To help ensure optimal performance, an administrator must upgrade the AIX OS, set the processor compatibility mode to the default, and migrate to IBM Power10 hardware by using LPM. This process enables full usage of all available features in the new hardware for the corresponding AIX level, without requiring extra operating system configuration or tuning.

Figure 8-4 shows setting the processor mode in your Hardware Management Console (HMC) LPAR profile.

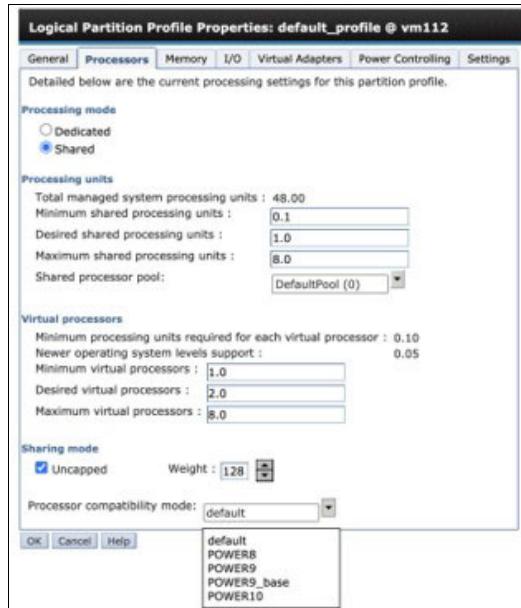


Figure 8-4 Processor modes in an HMC profile

Note: For more information about processor compatibility modes, see [this IBM document](#).

8.5 AIX and Power10 support

AIX is continuously enhanced to support new features that are introduced in IBM POWER processors. AIX 7.3 provides full support for the capabilities that are available in Power10. When the next generation of IBM POWER processors is announced, AIX will be updated to support the new functions that are introduced by that hardware.

Figure 8-5 on page 251 shows how AIX 7.3 supports Power10 features.

Introducing: AIX 7.3 and software enhancements for the Power10 family

4Q Software Announce: 10/19/2021
4Q Software GA: 12/10/2021



AIX
 IBM PowerHA IBM PowerVM
 IBM PowerVC IBM PowerSC
 VM Recovery Manager

Respond faster to business demands	<ul style="list-style-type: none"> Extends compute scaling to the full capabilities of Power10 128TB file and filesystem capacities for growing data needs Integrates use of on-chip NX GZIP with AIX commands and libs Out-of-the-box ready for Ansible with Python 3
Protect data from core to cloud	<ul style="list-style-type: none"> New IP security protects data in motion (IKEV2, NAT-T) PowerSC and PowerHA now include MFA On-chip acceleration for logical volume encryption of <code>rootvg</code> PowerHA supports secure encrypted data within the cluster
Streamline insights and automation	<ul style="list-style-type: none"> AI inferencing at the point of data with in-core MMA ESSL 7.1 support with AIX 7.3 and OpenXL C/C++ compilers RHEL or OpenShift "side-cars" for Enterprise AI with AIX AI inferencing with python and other opensource packages
Maximize availability and reliability	<ul style="list-style-type: none"> Enablement for LKU Live Library Update Enhancements Cross-cluster verification for quicker consistent deployment Multi-node PowerVC for improved resilience and scalability PowerHA/Spectrum Virtualize backup/recovery to IBM Cloud Faster DLPAR operations

Figure 8-5 AIX 7.3 support for Power10 features

This support consists of the following enhancements.

- ▶ Responds faster to business demands:
 - Adds Python 3.9.6. The new command to start Python is `/usr/bin/python3`. The version of Python that is included with AIX 7.3 works with Ansible solutions.
 - Extends the scalability of AIX, supporting a maximum of 240 cores (1920 hardware threads) in a single Power10 LPAR.
 - The JFS2 file system size and file size limits are increased beyond 32 TB and 16 TB.
 - Supports the usage of the on-chip NX GZIP accelerator in Power10 and Power9 servers. The `pigz` (parallel gzip) open source command and the AIX zlibNX library are included in the AIX 7.3 default installation. Both the `pigz` command and the zlibNX library transparently use the NX GZIP compression accelerator.
- ▶ Protect data from core to cloud:
 - IPsec support for IKE fragmentation with IKEv2.
 - IPsec support for NAT-T with IKEv2.
 - Provides enhanced support for logical volume (LV) encryption to include encryption for LV support in `rootvg` and dump device.
 - PowerSC and PowerHA now include multi-factor authentication (MFA).
 - PowerHA supports encrypted data volumes within the cluster.
- ▶ Streamline insights and automation:
 - AIX 7.3 can use the Matrix Math Accelerator (MMA) AI Accelerator instructions by using Power10 processor compatibility mode.
 - IBM Open XL C/C++ for AIX is a standards-based, high-performance compiler that supports the development and maintenance of C and C++ applications for IBM Power servers. It generates code that is optimized for the Power10 architecture, enabling efficient hardware usage. This release introduces several new built-in functions that unlock Power10 specific instructions.
 - Several open-source packages such as PyTorch and NumPy were tested on AIX and can be used for AI use cases that enable inferencing at the point of data.

- Applications on AIX can call out to Red Hat Enterprise Linux (RHEL) or Red Hat OpenShift “side-cars” to use the MMA accelerated AI capability for inferencing.
- RHEL or Red Hat OpenShift “side-cars” for Enterprise AI can also use relational databases on AIX to feed models or store predictions from inferencing.
- ▶ Maximize availability and reliability:
 - Reduces the amount of time that is required to dynamically add processor and memory resources to a running LPAR.
 - Reduces IPL times for multi-terabyte memory LPARs.
 - LKU supports the changing of selected boot time parameters without a requirement for a restart.
 - SMB 3.0.2 support for LKU.

8.6 AIX support for open source

Support for open-source software on AIX began with the release of AIX Version 5 in 2002. This version, which is known as *AIX5L*, marked the growing integration of AIX with Linux tools and utilities, recognizing their increasing popularity and practical value. In 2006, IBM began offering these Linux tools and utilities on separately orderable media, branded as the *AIX Toolbox for Open Source Software*. This package included both executable and source versions of various open-source applications.

The AIX Toolbox for Open Source Software is a curated collection of open-source and GNU software that is built specifically for IBM AIX systems. It provides a robust development environment that is used by many Linux application developers. The tools are packaged in the RPM format. Because of the strong synergy between Linux and AIX, and the long history of AIX standards compliance, rebuilding open-source applications for AIX is relatively straightforward. This toolkit highlights the deep connection between the Linux and AIX OSs.

8.6.1 AIX Toolbox for Open Source Software

The AIX Toolbox for Open Source Software can be downloaded from [AIX Toolbox for Open Source Software](#), or you can obtain an ISO image from [Entitled Systems Support \(ESS\)](#).

To install the toolbox repository and make `dnf` available on AIX, see “Creating an AIX image in PowerVC that includes the AIX Toolbox for Open Source Software” on page 256.

The bash shell

Start with the bash shell. The bash runtime environment is built into the AIX 7.3 runtime environment, as shown in Example 8-1.

Example 8-1 The bash shell that is built in to AIX 7.3

```
root@rbknim:/updates>lslpp -l | grep bash
  bash.rte                  5.2.15.0  COMMITTED  bash shell
  bash.rte                  5.2.15.0  COMMITTED  bash shell
root@rbknim:/updates>
```

You can switch to a bash shell by running `bash`, as shown in Example 8-2 on page 253.

Example 8-2 Changing to a bash shell

```
root@rbknim:/>bash  
bash-5.2#  
bash-5.2#  
(reverse-i-search)`':  
bash-5.2#
```

Matrix Math Accelerator in OpenBLAS on IBM AIX

OpenBLAS is a widely used open-source BLAS library that is favored by the scientific community. It accelerates linear algebra computations through low-level routines that operate on vectors and matrices by using platform-specific optimizations.

Different levels of BLAS routines were optimized to use MMA when running on Power10 processor-based systems with AIX (big endian) and Linux (little endian). The OpenBLAS application programming interfaces (APIs) were modified internally to use MMA C data types, functions, and instructions on Power10. To benefit from MMA when running AI workloads, users download the MMA-optimized OpenBLAS package from the AIX Toolbox for Open Source Software on a Power10 server and run their AI programs without modification.

Figure 8-6 shows a list of recent additions to the AIX Toolbox for Open Source Software.

New Areas of Exploration	
• Gcc 10.X/11.X with POWER10 support	• LLVM
• OpenBLAS integration with POWER10 MMA	• Rust
• MMA optimized Python ML and inferencing stacks	
Recent New Packages	
• tripwire	• SWIG
• rkhunter	• augeas
• tcpping	• cmocka
• Etcd	• lutok
• pv (pipe viewer)	• libgit2
	• pigz
	• atf
	• tmux
	• ninja-build
	• Libgit2
	• dnf
Security Related Updates	
• >400 CVEs reviewed in recent months, roughly 70 packages updated due security vulnerabilities	

Figure 8-6 Additions to the AIX Toolbox for Open Source Software

The AIX Toolbox for Open Source Software now includes a new version of **gcc** that supports Power10 instructions, including MMA. The integration of the OpenBLAS library with Power10 MMA on AIX 7.3 provides a strong foundation for running open-source machine learning (ML) and inferencing frameworks, such as PyTorch, NumPy, and SciPy, with Power10 acceleration on AIX.

Note: For more information about the MMA in OpenBLAS on IBM AIX, see [OpenBLAS on AIX](#).

Ansible collections for AIX and IBM Power

The IBM Power AIX collection provides Ansible modules to help administrators automate workloads and tasks on IBM Power servers. This collection can be found at [IBM Power Systems AIX Collection for Ansible](#).

For more information about the use of Ansible in the IBM Power environment, including support for AIX, see *Using Ansible for Automation in IBM Power Environments*, SG24-8551.

Here are some examples of using this collection in periodic tasks:

- ▶ Upgrading the VIOS, and managing file systems that are provided in the `ibm.power_vios` collection at [this GitHub repository](#).
- ▶ You can use the Ansible `ibm.power_hmc` collection to provision or modify an LPAR through the HMC. The collection also supports other HMC commands for gathering information about managed LPARs.
- ▶ Management of AIX updates is provided. Using Ansible can simplify the maintenance of security updates, which saves time and helps ensure compliance with your update strategy.

Ansible collections are easily installed on an Ansible workstation by using the `ansible-galaxy` command, as shown in Example 8-3.

Example 8-3 Ansible-galaxy collection installation

```
$ ansible-galaxy collection install ibm.power_hmc
Process install dependency map
Starting collection install process
Installing 'ibm.power_hmc:1.5.0' to
'/home/admin/.ansible/collections/ansible_collections/ibm/power_hmc'
```

When automating VM deployment on IBM Power servers without an existing OpenStack environment that is provided by PowerVC, administrators can use Ansible to automate the creation of virtual Fibre Channel (FC) adapters. This automation generates the required LPAR profile and WWPNs, preparing the VM for storage area network (SAN) Logical Unit Number (LUN) assignment by the storage administrator. During the boot process, the VM can detect the zoned LUNs and access its assigned storage.

Alternatively, this Ansible job can be integrated into a broader workflow that includes automated SAN tasks to provision the required LUNs. Ansible bridges the gap by automating the configuration of virtual Fibre Channel (FC) adapters, enabling seamless integration with SAN storage provisioning, even in environments without a full PowerVC OpenStack deployment.

Example 8-4 uses the `npiv_config` option of the `powervm_lpar_instance` module to create two virtual FC adapters on each VIOS for a VM LPAR profile that is named `ansi_test`. The HMC password is passed by using an Ansible vault

Example 8-4 Creating an LPAR with virtual FC adapters

```
$ cat hmc_vmcreation.yml
---
- name: HMC create and activate logical partition
  hosts: hmc
  gather_facts: no
  collections:
    - ibm.power_hmc
  connection: local
  vars:
    curr_hmc_auth:
      username: hscroot
      password: !vault |
        $ANSIBLE_VAULT;1.1;AES256
```

```

61323366356163383139373862366264613032356362353363532633031376637663463386132
3137653531643563383630346539363863613563393364350a633332613534366366303865323665
39343239373134393731626130646635663765386566613462643531323339646538303639663038
3933366261663034300a333262626532633261643231303565383632353830386666623730646666
3532

tasks:
- name: Create an AIX/Linux logical partition
  powervm_lpar_instance:
    hmc_host: '{{ inventory_hostname }}'
    hmc_auth: "{{ curr_hmc_auth }}"
    system_name: xxxxxxxxxxxx
    vm_name: ansi_test
    proc: 1
    proc_unit: 0.5
    mem: 4096
    max_mem: 8192
    min_mem: 2048
    virt_network_config:
      - network_name: VLAN100-ETHERNET0
    npiv_config:
      - vios_name: VIOS01
        fc_port: fcs0
      - vios_name: VIOS02
        fc_port: fcs1
      - vios_name: VIOS01
        fc_port: fcs0
      - vios_name: VIOS02
        fc_port: fcs1
    os_type: aix_linux
    state: present
    register: ansi_testout

- name: print the stdout of the lpar
  debug:
    msg: '{{ ansi_testout }}'

- name: Activate the created lpar
  powervm_lpar_instance:
    hmc_host: '{{ inventory_hostname }}'
    hmc_auth: "{{ curr_hmc_auth }}"
    system_name: xxxxxxxxxxxxxxxx
    vm_name: ansi_test
    keylock: normal
    action: poweron

```

To see what is available for IBM Power and AIX, see the [Ansible Galaxy site](#).

Note: For more information about the Ansible HMC modules, see [ibm.power_hmc](#).

Note: For more information about AIX Toolbox for Open Source Software, see [AIX Toolbox for Open Source Software: What's New?](#)

As a best practice, use DNF (the next-generation replacement for YUM) to install and manage open-source software (OSS) packages and dependencies from the AIX Toolbox for Open Source Software.

Note: For more information about installing DNF for the AIX Toolbox for Open Source Software, see this [IBM blog](#).

Creating an AIX image in PowerVC that includes the AIX Toolbox for Open Source Software

This section describes how to use IBM PowerVC to deploy a VM with the AIX Toolbox for Open Source Software. Start by creating an image to deploy AIX 7.3 by completing the following steps:

1. Create a LUN that is the size of the required rootvg, as shown in Figure 8-7.

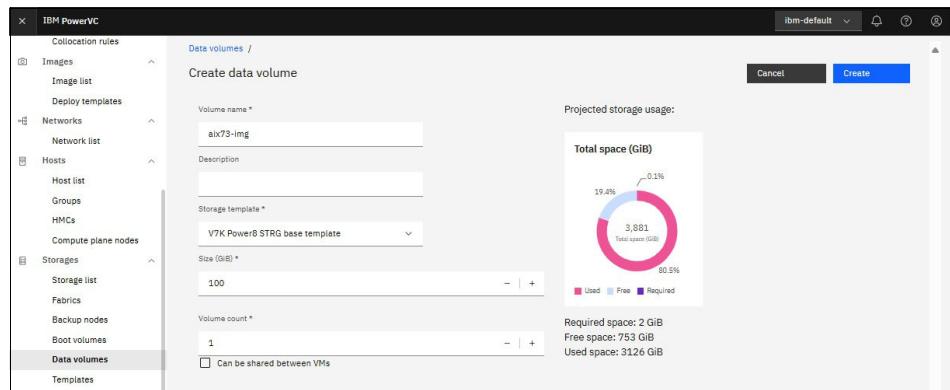


Figure 8-7 Creating a LUN that is the size of the required rootvg

2. Create a blank image for that LUN, as shown in Figure 8-8.

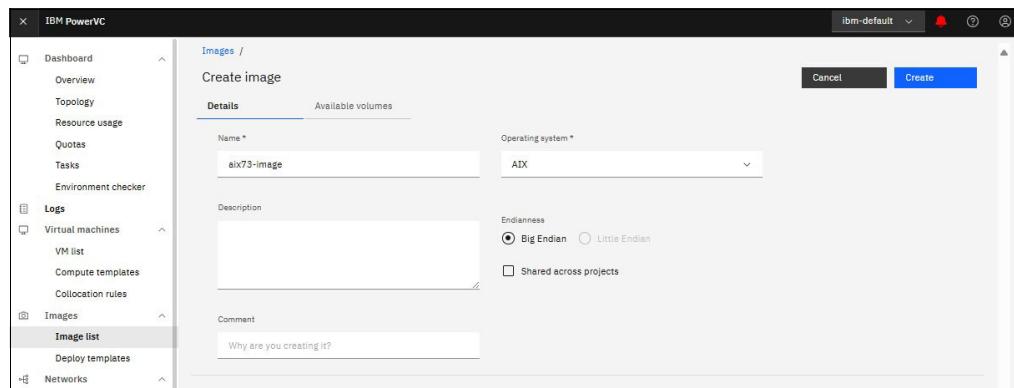


Figure 8-8 Creating an image

3. Add the blank LUN to that image. Add 0 to for the boot order to define it as a boot disk, as shown in Figure 8-9 on page 257.

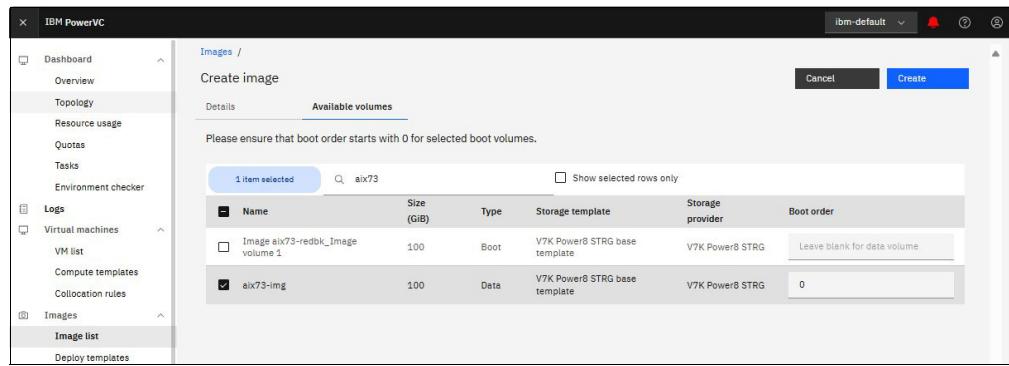


Figure 8-9 Adding the blank LUN to that image

4. Deploy a VM for an OS installation.

After deploying a VM with the image, the VM will not contain an OS. This step initiates a boot to System Management Services (SMS), allowing the administrator to attach an OS disk for installation. At this stage, the VM can be assigned any name because it might be deleted later. Once the OS is installed, the VM can be captured and used as a base image for deploying more VMs. The original VM may be retained for change management and future recapture, or deleted if no longer needed.

5. Assign a network to the VM and identify the primary network, as shown in Figure 8-10.

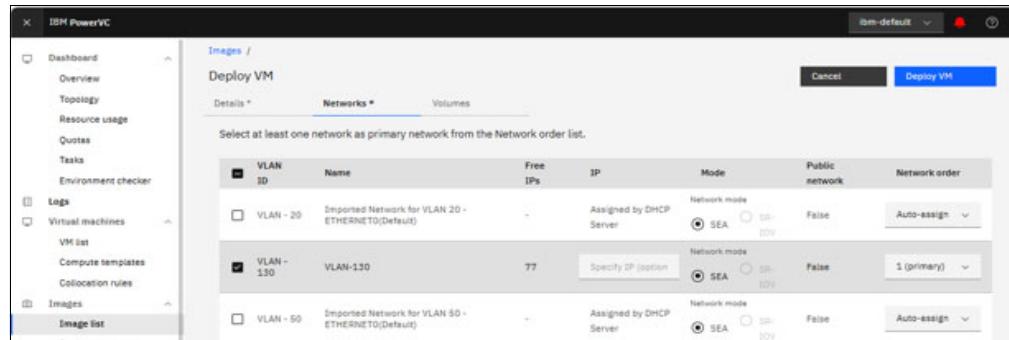


Figure 8-10 Assigning a network to the VM

6. Deploy the blank VM, as shown in Figure 8-11.

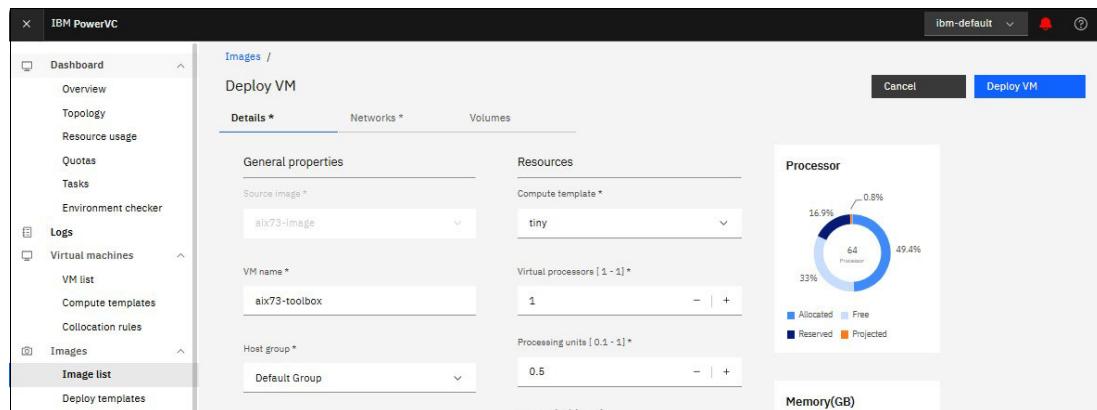


Figure 8-11 Deploying the blank VM

7. You see the VM building. Wait until an IP address is assigned to help ensure that it is available for login, as shown in Figure 8-12.

Virtual machines									
<input type="button"/> Search									
Name	Host	IP	State	Health	Resources	Description	VM owner	Expiration date	Running task
vm01	E-		Building	Pending	4 vCPU / 0 GB / 0 PU				<input type="button"/> Building
Items per page: 20 1–1 of 1 items									

Figure 8-12 VM building

When the VM is built, it is active and an IP address is assigned, as shown in Figure 8-13 (the IP address is partially obscured for security).

Name	Host	IP	State	Health	Resources
aix73-toolbox	Server-9119-MME	3.130.95	Active	Warning	1 vCPU / 4 GB / 0.5 PU

Figure 8-13 VM built and active

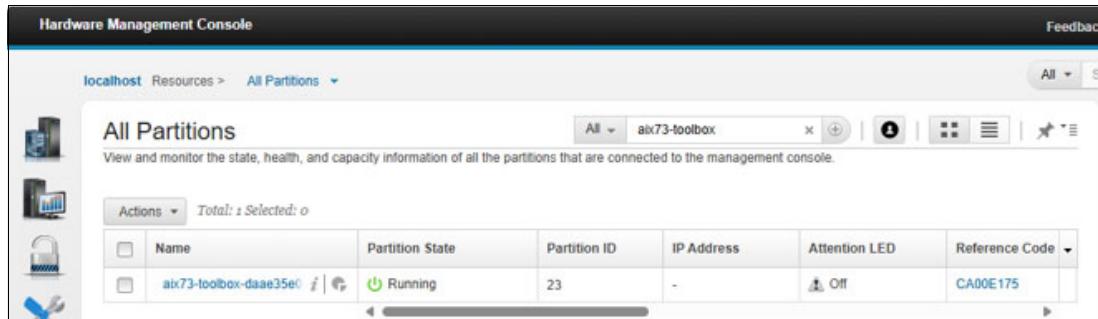
8. Run `ssh` in to the HMC, run the `vtmenu` command, select the managed system, and find the blank VM name. The console opens. Your VM will be attempting to restart from a blank disk, as shown in Figure 8-14.

```
No OS image was detected by firmware.  
At least one disk in the bootlist was not found yet.  
Firmware is now retrying the entries in the bootlist.  
Press ctrl-C to stop retrying.  
  
BOOTP Parameters:  
-----  
chosen-network-type = ethernet,auto,none,auto  
server IP = 0.0.0.0  
client IP = 0.0.0.0  
gateway IP = 0.0.0.0  
device = /vdevice/1-1an@30000020  
MAC address = fa 1b 40 52 3a 20  
loc-code = U9119.MME.21BE747-V23-C32-T1  
  
BOOTP request retry attempt: 1  
BOOTP request retry attempt: 2  
BOOTP request retry attempt: 3  
BOOTP request retry attempt: 4
```

Figure 8-14 BOOTP retries from a blank disk

9. Switch to the VIOS for the managed system. Log in to each of the VIOS partitions. Check the existence of the VM in the HMC and get the partition ID.

Figure 8-15 shows finding the partition ID of an LPAR.



Name	Partition State	Partition ID	IP Address	Attention LED	Reference Code
aix73-toolbox-daae35e0	Running	23	-	Off	CA00E175

Figure 8-15 Identifying the partition ID

10. Convert the partition ID to HEX, which in this case is 17. Search for the virtual adapter in the VIOS by using the **lsmmap** command, as shown in Example 8-5.

Example 8-5 VIOS searches for the hex ID of the LPAR

```
$ lsmmap -all | grep 00017
vhost17          U9119.MME.xxxxx-V1-C72           0x00000017
```

11. Create a file-backed optical device (vtopt18) to mount an ISO for the blank VM. Do not use the existing vtopt that is used by PowerVC to create the blank VM. Create one by running the VIOS **mkvdev** command, as shown in Example 8-6.

Example 8-6 Creating a virtual optical device

```
$ lsmmap -vadapter vhost17
SVSA          Physloc          Client Partition ID
-----
vhost17        U9119.MME.21BE747-V1-C72        0x00000017
VTD            vtopt17
Status          Available
LUN             0x8100000000000000
Backing device  /var/vio/VMLibrary/vopt_daae35e0a4074dc8894a7bebe2f39d19
Physloc
Mirrored       N/A
$
$ mkvdev -fbo -vadapter vhost17
vtopt18 Available
$ lsmmap -vadapter vhost17
SVSA          Physloc          Client Partition ID
-----
vhost17        U9119.MME.21BE747-V1-C72        0x00000017
VTD            vtopt17
Status          Available
LUN             0x8100000000000000
Backing device  /var/vio/VMLibrary/vopt_daae35e0a4074dc8894a7bebe2f39d19
Physloc
Mirrored       N/A
VTD            vtopt18
Status          Available
LUN             0x8200000000000000
Backing device
```

Physloc	
Mirrored	N/A

Note: For more information about creating images for PowerVC, see [Working with images](#).

12. To mount an AIX ISO, the image must first be uploaded to the VIOS virtual optical repository. You can do this task by using either the CLI or the HMC.
- To do this task from the CLI, see the [Creating a Virtual Optical Repository on the VIOS using the command line](#).
 - To do this task from the HMC, see [Creating the Virtual Optical Repository n the VIOS from the HMC](#).
- Copy the ISO to the VIOS and load it into the VIOS virtual optical library by using the VIOS **mkvopt** command.
13. You can see the newly available ISO and other existing RHEL8.4 ISOs that are mounted and being used by other VMs, as shown in Example 8-7.

Example 8-7 Creating the optical device displaying the mounted volumes

```
$ mkvopt -name AIX_v7.3_7300-00-00-2147_DVD_1 -file
/home/padmin/AIX_v7.3_Install_7300-00-00-2147_DVD_1_of_2_122021_LCD8265100.iso
$ lsrep
Size(mb) Free(mb) Parent Pool          Parent Size      Parent Free
25496     13642  rootvg                739328         591872

Name                               File  Size  Optical
Access
AIX_v7.3_7300-00-00-2147_DVD_1           3726  None   rw
RHEL8.4                           8128  vtopt0  ro
RHEL8.4                           8128  vtopt1  ro
RHEL8.4                           8128  vtopt2  ro
```

14. Once the media repository and AIX ISO are available, the image can be loaded into vtopt18 by using the **loadopt** command. After loading, mount the AIX 7.3 ISO to the new VM, as shown in Example 8-8.

Example 8-8 Mounting the ISO

```
$ loadopt -vtd vtopt18 -disk AIX_v7.3_7300-00-00-2147_DVD_1
$ lsmap -vadapter vhost17
SVSA          Physloc          Client Partition ID
-----
vhost17      U9119.MME.21BE747-V1-C72      0x00000017

VTD          vtopt17
Status       Available
LUN          0x8100000000000000
Backing device /var/vio/VMLibrary/vopt_daae35e0a4074dc8894a7bebe2f39d19
Physloc
Mirrored    N/A

VTD          vtopt18
Status       Available
LUN          0x8200000000000000
Backing device /var/vio/VMLibrary/AIX_v7.3_7300-00-00-2147_DVD_1
```

Physloc	
Mirrored	N/A

15. Restart the VM and install the OS from the SMS menu. Start SMS. For more information, see [Example Using SMS To Choose Boot Device](#).
16. Install AIX by using the BOS menu (SMS). For more information, see [Using BOS menus](#).
17. Configure a network to copy the AIX Toolbox for Open Source Software ISO to the VM.

To include the AIX Toolbox for Open Source Software in every AIX VM at deployment time, add the toolbox to this base image VM.

Installing DNF and the AIX Toolbox for Open Source Software repository

To install DNF and the IBM AIX Toolbox for Open Source Software repository, see [this blog](#).

The definition for the remote DNF repository is in the /opt/freeware/etc/dnf/dnf.conf file. This file references the /mnt location where your ISO image is mounted. You can either retain the ISO within the AIX image to deploy it in a file system (helps ensure that it is copied to each new VM) or update the /opt/freeware/etc/dnf/dnf.conf file to point to a remote YUM repository within your organization. If you choose to retain the ISO, you must mount it on /mnt each time that you need to use it.

You can create a new YUM repository server by deploying a new RHEL VM by using PowerVC.

To install DNF, complete the following steps:

1. On the YUM repository server, run the commands that are shown in Example 8-9.

Example 8-9 Installing the YUM repository server

```
# dnf install createrepo
# dnf install http
# scp
root@xx.xx.xx.xx:/updates/ESD-Toolbox_for_Linux_Apps_Common_7.2-7.3_122024_LCD410740.iso .
# mount -t iso9660 -o loop
/root/ESD-Toolbox_for_Linux_Apps_Common_7.2-7.3_122024_LCD4107740.iso /mnt
# cd /var/www/html
# cp -rp /mnt/RPMS /var/www/html
# createrepo --database /var/www/html/RPMS
Directory walk started
Directory walk done - 1457 packages
Temporary output repo path: /var/www/html/RPMS/.repodata/
Preparing sqlite DBs
Pool started (with 5 workers)
Pool finished
```

2. Copy the dnf.conf file to /etc/yum/yum.repos.d/aix-toolbox.repo <filename> on the YUM repository server. Update the contents to point to your Yum server, as shown in Example 8-10.

Example 8-10 The dnf.conf file

```
[main]
cachedir=/var/cache/dnf
keepcache=1
debuglevel=2
```

```

logfile=/var/log/dnf.log
exactarch=1
gpgcheck=1
installonly_limit=3
clean_requirements_on_remove=True
best=True

plugins=1

[AIX_Toolbox]
name=AIX generic repository
baseurl=http://yum-server/RPMS/ppc/
enabled=1
gpgcheck=0

[AIX_Toolbox_noarch]
name=AIX noarch repository
baseurl=http://yum-server/RPMS/noarch/
enabled=1
gpgcheck=0

[AIX_Toolbox_73]
name=AIX 7.3 specific repository
baseurl=http://yum-server/RPMS/ppc-7.3/
enabled=1
gpgcheck=0

```

3. Start and enable httpd on the YUM repository server, as shown in Example 8-11.

Example 8-11 Starting httpd on the YUM repository server

```

# firewall-cmd --permanent --add-service=http
# firewall-cmd --permanent --add-port=80/tcp
# firewall-cmd --reload
# systemctl start httpd
# systemctl enable httpd

```

4. Copy the updated aix-toolbox.repo file to the /opt/freeware/etc/dnf/dnf.repo file on the base client before capture.
5. Ensure that the yum-server hostname is resolvable in /etc/hosts or a DNS server. If it is a DNS server, ensure that the DNS server is defined in the base VM.

Now, you have a centralized YUM repository server to serve all the standard VMs that you deploy from PowerVC. You no longer need to mount the toolbox ISO locally.

6. Install cloud-init on your base AIX VM by using dnf, as shown in Example 8-12.

Example 8-12 Installing cloud-init

```
# dnf install cloud-init
```

Now, your AIX VM may be captured as an OpenStack image. All future VMs that are deployed from PowerVC by using this image are enabled for capture.

You do not need any additional configuration changes or customizations for cloud-init. Although further customization is possible, it is not required for creating a base image deployment.

When done, remove the TCP/IP interface from the VM before capture, as shown in Example 8-13.

Example 8-13 Removing the network interface

```
# ifconfig en0 down detach
```

7. In PowerVC, capture the VM. You are prompted to verify that you prepared the VM. The minimum requirement is that cloud-init is installed.

Confirm that the VM is prepared, update the image name if required, and select **Capture**, as shown in Figure 8-16.

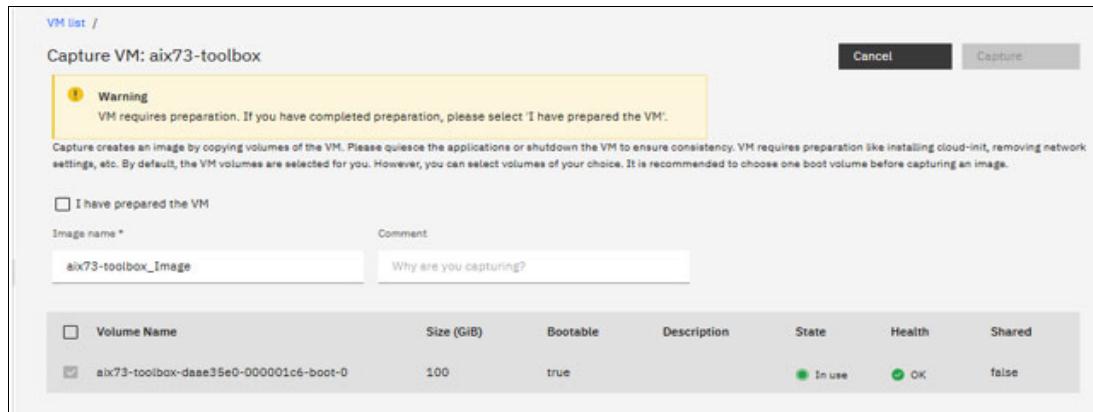


Figure 8-16 Capturing the VM in PowerVC

Now, you have an image that can be deployed into a running VM with a hostname that is the same as the VM name that is specified in PowerVC. You can log in to this VM by using the IP address that is assigned by PowerVC.

8. Deploy a VM.

Go to images in PowerVC and deploy an instance of the image that you created. Name it mod1. PowerVC should deploy a VM with the hostname mod1.

When the image is prepared and captured, you may deploy a base AIX 7.3 VM in less than 10 minutes from PowerVC, and it includes the IBM AIX Toolbox for Open Source Software.

8.7 Live Update

AIX 7.2 introduces the AIX Live Update capability, which is a feature that is designed to reduce workload downtime during kernel patching. Previously, deploying kernel fixes required a system restart, which interrupted running applications. In contrast, Live Update applies interim fixes without stopping active workloads, allowing them to immediately use the updated code.

IBM delivers kernel patches as *interim fixes*. In AIX versions before Version 7.2, modifying the AIX kernel or persistently loaded kernel extensions required a host LPAR restart. Although AIX 7.1 and earlier supported concurrent update-enabled interim fixes for a limited set of kernel changes, this approach had restrictions. AIX 7.2 extends the principles of concurrent updates through Live Update, enabling a broader range of kernel fixes to be applied without restarting the system.

8.7.1 Example use of Live Update

For this example, you continue with the installed AIX VM mod1 at oslevel 7300-00-00-0000.

When you start the VM, the console output shows that DNF is installed. However, some modules are not loading correctly, as shown in Figure 8-17.

```
0513-059 The inetd Subsystem has been started. Subsystem PID is 3998104.  
0513-059 The snmpd Subsystem has been started. Subsystem PID is 5898688.  
0513-059 The hostmibd Subsystem has been started. Subsystem PID is 4981172.  
0513-059 The snmpmibd Subsystem has been started. Subsystem PID is 4915662.  
0513-059 The aixmibd Subsystem has been started. Subsystem PID is 5767610.  
Finished starting tcppip daemons.  
Starting NFS services:  
0513-059 The biod Subsystem has been started. Subsystem PID is 5046704.  
exec(): 0509-036 Cannot load program /usr/lib/sendmail because of the following errors:  
    0509-022 Cannot load module /usr/lib/sasl/libntlm.so(shr.o).  
    0509-150  Dependent module libcrypto_compat.a(libcrypto.so) could not be loaded.  
    0509-022 Cannot load module libcrypto_compat.a(libcrypto.so).  
    0509-026 System error: A file or directory in the path name does not exist.  
    0509-022 Cannot load module sendmail.  
    0509-150  Dependent module /usr/lib/sasl/libntlm.so(shr.o) could not be loaded.  
exec(): 0509-036 Cannot load program /usr/lib/sendmail because of the following errors:  
    0509-022 Cannot load module /usr/lib/sasl/libntlm.so(shr.o).  
    0509-150  Dependent module libcrypto_compat.a(libcrypto.so) could not be loaded.  
    0509-022 Cannot load module libcrypto_compat.a(libcrypto.so).  
    0509-026 System error: A file or directory in the path name does not exist.  
    0509-022 Cannot load module sendmail.  
    0509-150  Dependent module /usr/lib/sasl/libntlm.so(shr.o) could not be loaded.  
exec(): 0509-036 Cannot load program /usr/lib/sendmail because of the following errors:  
    0509-022 Cannot load module /usr/lib/sasl/libntlm.so(shr.o).  
    0509-150  Dependent module libcrypto_compat.a(libcrypto.so) could not be loaded.  
    0509-022 Cannot load module libcrypto_compat.a(libcrypto.so).  
    0509-026 System error: A file or directory in the path name does not exist.  
    0509-022 Cannot load module sendmail.  
    0509-150  Dependent module /usr/lib/sasl/libntlm.so(shr.o) could not be loaded.  
0513-059 The rpc.lockd Subsystem has been started. Subsystem PID is 8913188.  
Completed NFS services.
```

Figure 8-17 A new AIX VM that is installed with DNF at 7300-00 exhibits library load errors

Note: The problem was identified and described in this [technote](#). The solution requires an AIX update.

This issue is fixed with an upgrade to AIX 7.3 TL0 SP3 or later, or TL1 SP0 or later.

However, the VM was handed over to the development teams, who were not aware of the issue. These teams waited a significant amount of time for the VM and did not want any further disruption that was caused by restarts.

Also, when creating the base image during the previous PowerVC deployment tasks, the default compute template was used, as shown in Figure 8-18 on page 265. This template restricts the VM to a maximum of one processor. You want to increase this limit without shutting down the VM.

Compute templates				
	Name	Processors	Processing units	Memory (GB)
<input type="checkbox"/>	customer	1	0.5	4
<input type="checkbox"/>	large	8	4	32
<input type="checkbox"/>	medium	4	2	16
<input type="checkbox"/>	ocp-control	4	1.5	32
<input type="checkbox"/>	ocp-worker	4	1	16
<input type="checkbox"/>	small	2	1	8
<input type="checkbox"/>	tiny	1	0.5	4
<input type="checkbox"/>	xlarge	16	8	64
<input type="checkbox"/>	xxlarge	32	16	128
Items per page:		9 ▾	1–9 of 9 items	1 ▾ of 1 page ▶

Figure 8-18 PowerVC Compute templates

Live Update can help you correct both problems.

8.7.2 Live Update concepts

In the AIX Live Update function, the LPAR where the operation begins is called the *original partition*. The process involves another LPAR, referred to as the *surrogate partition*. Checkpointing a workload involves freezing a running process and saving its state. Migrating checkpointed processes from one LPAR to another is known as *mobility*.

The AIX Live Update process is configured by modifying stanzas in the /var/adm/ras/liveupdate/lvupdate.file, which is derived from the lvupdate.template in the same directory. This update process runs as a singleton. The geninstall command creates the /usr/lpp/.genlib.lock.check lock file to help ensure that only one instance of the process is active at a time.

The Live Update operation runs in one of the following modes:

- ▶ Preview mode

Preview mode provides an estimate of the total operation time, application blackout duration, and the required resources, such as storage and memory, which are available to the user. These estimates assume that the surrogate partition has the same resources as the original partition. All provided inputs are validated, and Live Update limitations are checked.

- ▶ Automated mode

In automated mode, a surrogate partition with the same capacity as the original partition is created. After the Live Update operation completes, the original partition is turned off and discarded.

For more information about Live Update, see [Live Update Concepts](#).

Live Update evolution

The Live Update process originally used the `emgr` and `installp` commands to enable non-disruptive updates for kernel, library, and emergency fixes. It is now available for Service Pack and Technology Level updates. Beginning with AIX 7.2, LKU supported only fix-level updates. Over time, it evolved to include support for Service Pack and Technology Level updates. Figure 8-19 illustrates this evolution.

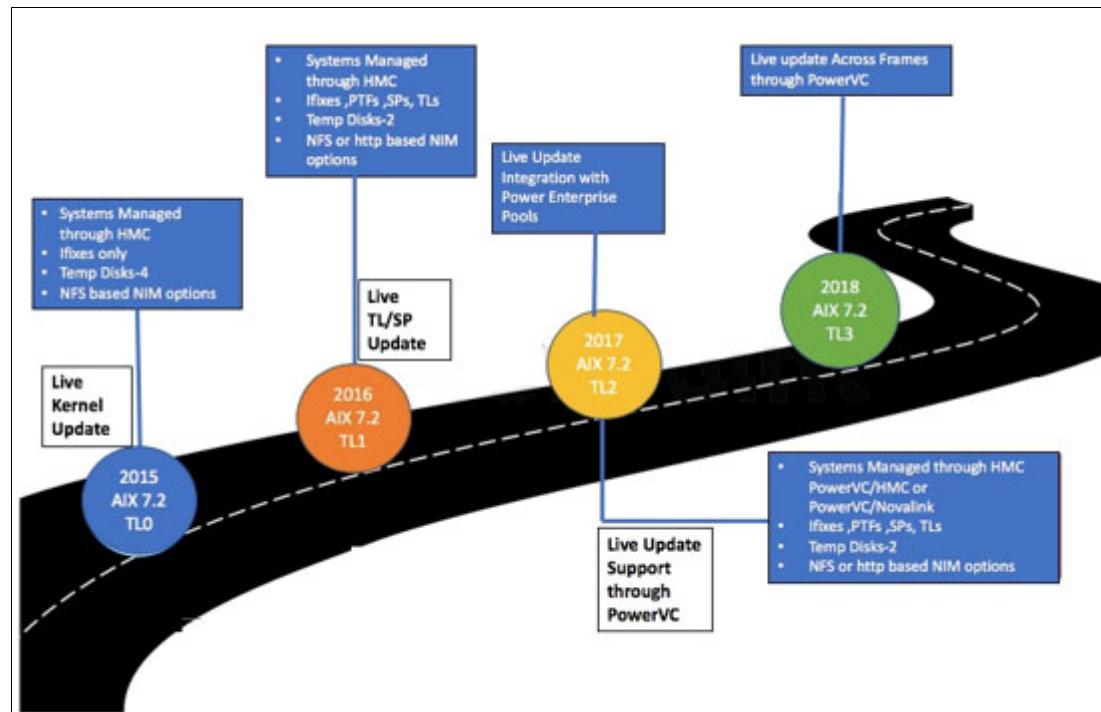


Figure 8-19 Live Update evolution¹

Live Update

Starting with AIX 7.2, the OS provides the AIX Live Update function. This capability is part of the AIX continuous availability features and eliminates downtime that is associated with restarts after applying specific interim fixes and kernel patches.

¹ Source: <https://www.ibm.com/support/pages/ibm-aix-72-live-kernel-update-reboot-free-world>

Concurrent AIX Update uses functional redirection within the in-memory image of the OS to fix corrected code. After a fix is identified, the corrected code is built, packaged, and tested according to the Concurrent AIX Update process. It is then delivered to the customer by using the existing interim fix package format.

Note: The ability to apply or remove a fix without requiring a restart is limited to Concurrent AIX Updates. Due to technological restrictions, some fixes cannot be delivered as Concurrent AIX Updates. In such cases, the fixes are provided as traditional interim fixes.

Traditional interim fixes for the kernel or kernel extensions still require a system restart for both activation and removal.

Live Kernel Update

The terms *Live Update* and *Live Kernel Update* are synonymous because both refer to the same process for applying interim fixes. The use of *kernel* emphasizes that a kernel-level interim fix typically requires a restart. In contrast, a fix that does not trigger a **bosboot** request also does not update the kernel and might not require a restart.

Live Library Update

The LLU function shifts applications from using an old library to an updated library without incurring downtime. This process is separate from Live Update or LKU operations.

A *library* is an entity that provides a set of variables and functions for use by a program. It can be an archive or a shared object file. On AIX, archives may contain both static object files and shared object files. In the LLU context, a library refers specifically to a shared object file contained within an archive.

The LLU function requires that the library is built as a split library. A library is considered split, or LLU-capable, when its shared object file is divided into two separate entities.

You can run the LLU operation by using the **11vupdate** command. A process is LLU-capable if at least one of its libraries can be dynamically replaced by using the **11vupdate** command. The command has the following limitations in this context:

- ▶ You cannot run the **11vupdate** command separately when a Live Update operation is in progress.
- ▶ You cannot run the LLU operation when the **11vupdate** command is running.

For more information about the LLU process, see this [IBM document](#).

Live Update restrictions and limitations

The following restrictions apply to Live Update:

- ▶ Do not change the rootvg, attached devices, or file systems. Do not run any volume group commands or restart the VIOS, or restart the PowerVC or HMC during the Live Update operation.
- ▶ After the Live Update operation completes, if only interim fixes were applied, the mdisk disk that is specified for the rootvg mirror volume group is labeled `old_rootvg`. This volume group can be used to restart the system and return to the previous version of the root volume group before the update was applied.
- ▶ Any existing `altinst_rootvg` label can cause the Live Update operation to fail.
- ▶ Live Update does not support encrypted disk volumes or file systems.
- ▶ Only supported storage providers in PowerVC are supported by a PowerVC Live Update procedure. Pluggable cinder drivers are not supported.

- ▶ IBM Power servers that are managed by PowerVM NovaLink cannot be used with VFC adapters on PowerVC.
- ▶ The Live Update operation does not support rootvg on PowerPath.
- ▶ The Oracle Real Application Cluster (RAC) database and IBM pureScale database are not supported by the Live Update operation.
- ▶ The Live Update feature is not supported on a partition that participates in Active Memory Sharing (AMS).
- ▶ The Live Update feature is not supported on a partition with the remote restart capability enabled. However, the Live Update feature is supported on a partition with the simplified version of the remote restart capability enabled.
- ▶ The console must be closed before running the Live Update operation. The Live Update operation fails if the console device is open for any process.
- ▶ Do not initiate an HMC-based Live Update operation on a partition that is managed by PowerVC. This type of operation can cause issues when PowerVC is actively managing partitions. To perform an HMC-based Live Update, unmanage the VM in PowerVC, complete the update, and then manage the VM again.

For more information and a full list of restrictions, see [Planning Restrictions](#).

8.7.3 Performing a Live Update by using Network Installation Manager

To perform a Live Update by using Network Installation Manager (NIM), complete the following steps:

1. Create the NIM server.

Create a NIM server at AIX TL 7300-03-00-2446. Mount the ISO for the same OS level and install the bos.sysmgt.nim.master file set, as shown in Example 8-14.

Example 8-14 Mounting the ISO for installation

```
#loopmount -i
/updates/AIX_v7.3_Install_7300-03-00-2446_DVD_1_of_2_122024_LCD8299201.iso -o "-V
udfs -o ro" -m /mnt
```

2. Install the bos.sysmgt.nim.master file set, as shown in Example 8-15.

Example 8-15 Installing the NIM master file set

```
# installp -gaXcd /mnt/installp/ppc/bos.sysmgt bos.sysmgt.nim.master
```

For an Ethernet network, use the command that is shown in Example 8-16 to set up the NIM server. This command creates a NIM server that uses the primary interface en0.

Example 8-16 Creating the NIM server

```
# nimconfig -a netname=master_net -a pif_name=en0 -a cable_type=tp -v -a
platform=chrp -a netboot_kernel=mp
```

3. Create the other resources for 1pp and spot, as shown in [this IBM document](#).
4. On the NIM server, install dsm.core from the mounted ISO by using the command that is shown in Example 8-17 on page 269.

Example 8-17 Installing dsm.core

```
# installp -gaXcd /mnt/installppc/dsm dsm.core
```

5. Create the NIM CEC resources for the connection to the HMC.
6. Create a user in the HMC that is named liveupdate, as shown in Figure 8-20.

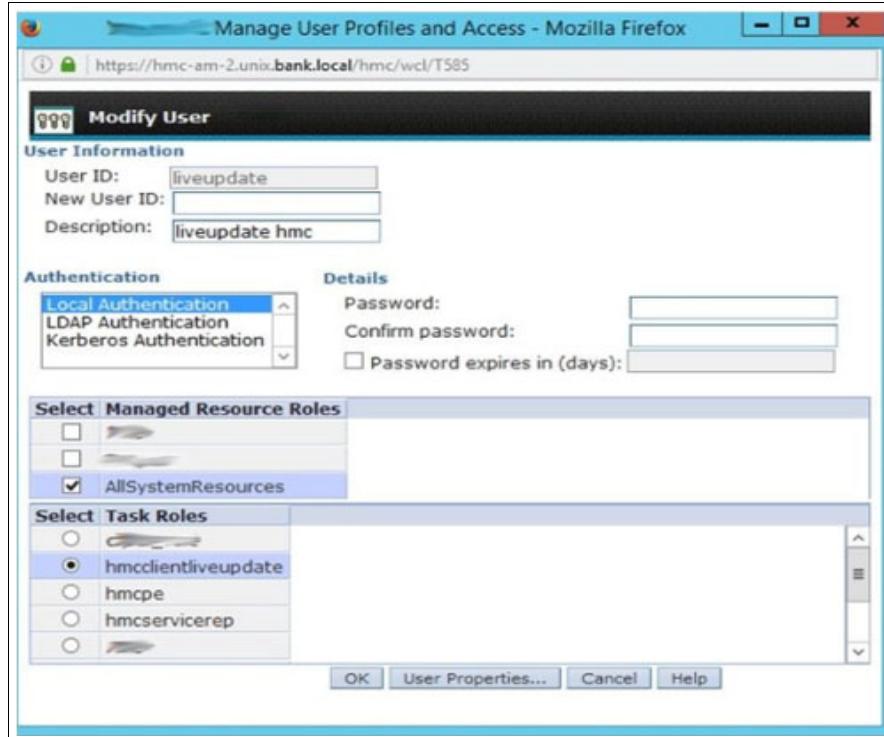


Figure 8-20 Creating a user that is named liveupdate

7. Create an encrypted password file on the NIM server for that user, as shown in Example 8-18.

Example 8-18 Creating an encrypted password file

```
# /usr/bin/dpasswd -f /export/nim/hmc_liveupdate_passwd -U liveupdate  
Password:  
Reenter password:  
Password file created.
```

8. Define that password file as a NIM object. Replace \$hmc-name with your DNS or hosts file that resolves to hmc <hostname>, as shown in Example 8-19.

Example 8-19 Defining a password file as an object

```
# nim -o define -t hmc -a ifl="find_net $hmc-name 0" -a net_definition="ent  
255.255.255.0" -a passwd_file=/export/nim/hmc_liveupdate_passwd $hmc-name
```

9. Now, place the hscroot and liveupdate users into the known_hosts file of the NIM server, you get the error message that is shown in Example 8-20 in the next step, which is shown in Example 8-21 to validate keys.

Example 8-20 Error message from validating a key without the correct entry in the known_hosts file

```
2760-287 [dkeyexch] Internal error - exchange script returns unknown error
```

Example 8-21 Login fails: Key not found

```
# ssh hscroot@$hmc-name
The authenticity of host 'xxxxxxxxxxxxxx' can't be established.
ED25519 key fingerprint is SHA256:P7x1z1Q+K5xPiuWs+XGcDQE++6cLMgFGR0kP81iKj+I.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

10. Do the key exchange between the NIM server and the HMC, as shown in Example 8-22.

Example 8-22 Exchanging keys

```
# dkeyexch -f /export/nim/hmc_liveupdate_passwd -I hmc -H $hmc-name
```

11. Authorize **ssh** between the NIM server and the HMC with hmcauth, as shown in Example 8-23.

Example 8-23 Authorizing ssh

```
# hmcauth -u <liveupdate-user> -p <password> -a <hostname-hmc>
# hmcauth -l
```

12. Define the IBM Power managed systems that are managed by the HMC as CEC objects in NIM, as shown in Example 8-24. Do this task for each managed system.

Example 8-24 Defining CEC objects

```
# nim -o define -t cec -a hw_type=9119 -a hw_model=MME -a hw_serial=xxxxxxx -a
mgmt_source=$hmc-name p8757
```

13. Create NIM definitions for the clients that undergo Live Update.

Use the **smit** menu so that you can add the LPAR ID as Identity, and Management Source as the managed system. Define the communication method as nimsh. These actions are shown in Figure 8-21.

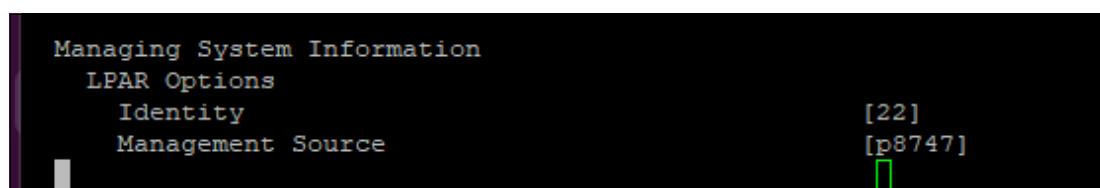


Figure 8-21 SMIT menu for adding LPAR definitions

14. Set up **nimsh** on the NIM server, and set up the client by using the commands that are listed in Example 8-25 on page 271 (the output from the commands is not captured).

Example 8-25 Setting up nimsh

```
On the NIM server small c
# nimconfig -c
On the NIM client big C
# niminit -a name=mod1 -a master=rbknim -a connect=nimsh
# nimclient -C
```

15. Test the connection from the NIM server by running **nim -o ls1pp mod1**, as shown in Example 8-26.

Example 8-26 Testing the NIM server connectivity by running a display command

```
root@rbknim:/>nim -o ls1pp mod1
      File set          Level  State       Description
-----
Path: /usr/lib/objrepos
    ICU4C.rte        7.3.0.0  COMMITTED  International Components for
                                Unicode
    Java8_64.jre     8.0.0.636  COMMITTED  Java SDK 64-bit Java Runtime
                                Environment
    Java8_64.sdk     8.0.0.636  COMMITTED  Java SDK 64-bit Development
```

16. Create a Live Update resource. Use **liveupdate.rte** for AIX 7.2 or later, as shown in Example 8-27.

Example 8-27 Creating a Live Update resource

```
root@mod1:/> ls1pp -L bos.liveupdate.rte
      File set          Level  State   Type  Description (Uninstaller)
-----
bos.liveupdate.rte      7.3.0.0  C      F      Live Update Runtime
```

17. The mod1 VM already has two available disks of the same size as rootvg, and there is enough spare CPU and memory capacity on the managed system. Copy **liveupdate.template** to **liveupdate.data**, as shown in Example 8-28.

Example 8-28 Copying the liveupdate template to liveupdate.data

```
# cp /var/adm/ras/liveupdate/lvupdate.template
/var/adm/ras/liveupdate/lvupdate.data
```

18. Update the contents with the required LPAR information, as shown in Example 8-29.

Example 8-29 Completing the LPAR information

```
general:
    mode = preview
    kext_check = no
disks:
    nhdisk = hdisk1
    mhdisk = hdisk2
    tohdisk =
    tshdisk =
hmc:
    lpar_id = 36
    management_console = xx.xx.xx.xx
    user = liveupdate
```

If you use the **geninstall -k** command for the Live Update, the **geninstall** command references this **lvupdate.data** file and the location of the emergency fix, as shown in Example 8-30.

Example 8-30 The geninstall command

```
# geninstall -k -p -d /tmp/cg/dummy dummy.150813.epkg.Z
```

However, with NIM, you can externalize the **lvupdate.data** file from the VM, which helps ensure that there are no details that are available for a local **geninstall** command on the VM.

19. Define the **lvupdate.data** as a NIM resource by copying the original **/var/adm/ras/liveupdate/lvupdate.template**, which is on the NIM server to **/export/fixes/lvupdate_data_<LPARNAME>** on the NIM server. Make the same changes for the client, except you do not need to add a mode when you use the NIM approach. Instead, the preview is managed by the **installp** preview flag when running the NIM operation.
20. Ensure that the value for **management_console** in the **live_update_data** file matches the value that is returned from the **hmcauth -l** command.

During the NIM Live Update task, the NIM-defined Live Update file is copied to the client's **/var/adm/ras/liveupdate/lvupdate.data** location. The **lpar_id** that is specified in the **lvupdate.data** file is a spare **lpar_id** that is used to create the surrogate LPAR, as shown in Example 8-31.

Example 8-31 The lvupdate.data file

general:

```
kext_check = no
```

disks:

```
nhdisk = hdisk1  
mhdisk = hdisk2  
tohdisk =  
tshdisk =
```

hmc:

```
lpar_id = 36  
management_console = xx.xx.xx.xx  
user = liveupdate
```

With this method, you have a file that is defined for the client on NIM, and can manage which clients are targeted for a Live Update process.

21. Define the NIM resource type **live_update_data**, as shown in Example 8-32.

Example 8-32 Defining the live_update_data resource on NIM

```
#nim -o define -t live_update_data -a server=master -a  
location=/export/fixes/liveupdate_data_<LPARname> liveupdate_data_<lpar-name>  
#lsnim -t live_update_data  
liveupdate_data_mod1 resources live_update_data
```

22. On the client VM mod1, you have three spare disks. Use **hdisk1** for a traditional **alt_disk_backup** in case the update process fails. The **hdisk2** is defined as the **ndisk**, so use it to create the surrogate VM. Then, use **hdisk3** to mirror the **rootvg** on the surrogate during the Live Update process. This process is shown in Example 8-33 on page 273.

Example 8-33 Showing the available hdisks

```
root@mod1:/tmp>lspv
hdisk0      00cbe757c4603ee5          rootvg      active
hdisk1      none                      None
hdisk2      none                      None
hdisk3      none                      None
root@mod1:/tmp>
```

23. Download and unpack this recent [interim fix](#) into a directory in the /export/efix location of the NIM server. Extract the file into a directory that is named kernext_fix, and check that it is concurrently enabled by running the command that is shown in Example 8-34.

Example 8-34 Downloading the interim fix

```
root@rbknim:/export/fixes/kernext_fix>ls *.epkg.Z
IJ52366s6a.241113.epkg.Z  IJ52533m8a.241204.epkg.Z  IJ52978m4a.241204.epkg.Z
IJ53001m3a.241216.epkg.Z
IJ52366s7a.241113.epkg.Z  IJ52610m2a.241204.epkg.Z  IJ52999m2a.241216.epkg.Z
IJ53001m4a.241204.epkg.Z
IJ52366s8a.241031.epkg.Z  IJ52977s2a.241113.epkg.Z  IJ52999m3a.241216.epkg.Z
IJ52421s1a.241112.epkg.Z  IJ52977s3a.241113.epkg.Z  IJ52999s4a.241105.epkg.Z
IJ52421s2a.241031.epkg.Z  IJ52977s4a.241031.epkg.Z  IJ53001m2a.241216.epkg.Z
root@rbknim:/export/fixes/kernext_fix>for fix in `ls *.epkg.Z` ; do emgr -d -e
$fix -v2 | grep "LU CAPABLE:" ; done
LU CAPABLE: yes
```

24. Define the interim fix on the NIM server as an lpp_source, as shown in Example 8-35.

Example 8-35 Defining the interim fix on the NIM server

```
# nim -o define -t lpp_source -a server=master -a
location=/export/fixes/kernext_fix kernext_fix

#lsnim -t lpp_source
aix_7_3_3_lpps    resources    lpp_source
aix_7_3_1_lpps    resources    lpp_source
kernext_fix       resources    lpp_source
```

Note: You might see the following warning:

warning: 0042-267 c_mk_lpp_source: The defined lpp_source does not have the "simages" attribute because one or more of the following packages are missing:

You can ignore this warning. It means that the lpp_source is not bootable because it was created from Service Pack or Technology Level upgrade media rather than AIX base media ISO. SPOT resource cannot be created from such a lpp_source.

For more information, see [How to run AIX update on NIM clients?](#)

25. For this Live Update, use one of the lpp_source files that were created on the NIM for AIX 7.3.1.

To change the max_cpu attribute of the profile, update the profile before the Live Update process. The original profile is shown in Figure 8-22.

General	Processors	Memory	I/O	Virtual Adapters	Power Controlling	Settings
Detailed below are the current processing settings for this partition profile.						
Processing mode						
<input type="radio"/> Dedicated <input checked="" type="radio"/> Shared						
Processing units						
Total managed system processing units : 32.00						
Minimum shared processing units : <input type="text" value="0.1"/>						
Desired shared processing units : <input type="text" value="0.5"/>						
Maximum shared processing units : <input type="text" value="1.0"/>						
Shared processor pool: <input type="text" value="DefaultPool (0)"/> ▾						

Figure 8-22 Changing the max_cpu initial view

The new value that you should see after the Live Update process completes is shown in Figure 8-23 on page 275.

General	Processors	Memory	I/O	Virtual Adapters	Power Controlling	Settings
Detailed below are the current processing settings for this partition profile.						
Processing mode						
<input type="radio"/> Dedicated <input checked="" type="radio"/> Shared						
Processing units						
Total managed system processing units : 32.00 Minimum shared processing units : <input type="text" value="0.1"/> Desired shared processing units : <input type="text" value="1"/> Maximum shared processing units : <input type="text" value="2"/> Shared processor pool: <input type="text" value="DefaultPool (0)"/> ▾						
Virtual processors						
Minimum processing units required for each virtual processor : 0.10 Newer operating system levels support : 0.05 Minimum virtual processors : <input type="text" value="1.0"/> Desired virtual processors : <input type="text" value="1.0"/> Maximum virtual processors : <input type="text" value="4"/>						

Figure 8-23 Max_cpu value that is expected after the change

Performing the Live Update

Run the Live Update by completing the following steps:

1. Run the Live Update process, You see a preview for the Technology Level Update, as shown in Example 8-36.

Example 8-36 Live Update preview

Results...

SUCCESSES

File sets that are listed in this section passed pre-installation verification and will be installed.

Mandatory File Set Updates

(being installed automatically due to their importance)
bos.rte.install 7.3.1.5 # LPP Install Commands

Prerequisites

(being installed automatically; required by file sets listed earlier)
bos.dsc 7.3.1.5 # Digital Signature Catalog

<< End of Success Section >>

```

+-----+
          BUILDDATE Verification ...
+-----+
Verifying build dates...done
FILESET STATISTICS
-----
 330 Selected to be installed, of which:
      2 Passed pre-installation verification
      328 Deferred (see *NOTE below)
-----
 2 Total to be installed

```

*NOTE The deferred file sets that are mentioned earlier will be processed after the installp update and its prerequisites are successfully installed.

The Live Update preview starts and provides an estimate of the time for the update. No update occurs now.

2. If you get a Live Update error, remove the live_update_<node name> nim resource by running the following command:

```
#nim -Fo remove liveupdate_data_<1par-name>
```

Change the live_update_<node name> file on the NIM server, and redefine the file to NIM by running the following command:

```
nim -o define -t live_update_data -a server=master -a
location=/export/fixes/liveupdate_data_<1par-name> liveupdate_data_<1par-name>
```

Run the preview again. The output should look like Example 8-37.

Example 8-37 Live Update preview successful

```

Verifying the environment...done
Verifying /var/adm/ras/liveupdate/lvupdate.data file...done
Computing the estimated time for the live update operation...done
Results...

```

EXECUTION INFORMATION

```

-----
  LPAR: mod1
  HMC: xx.xx.xx.xx
  user: liveupdate

```

```

Blackout time(in seconds): 10
Total operation time(in seconds): 806

```

```
<< End of Information Section >>
```

```

+-----+
          Live Update Requirement Verification...
+-----+

```

INFORMATION

```

-----
INFO: Any system dumps present in the current dump logical volumes will not be
available after the Live Update is complete.

```

```
<< End of Information Section >>
```

```

+-----+
          Live Update Preview Summary...
+-----+

```

```
+-----+
The live update preview succeeded.
*****
End of Live Update PREVIEW: No Live Update operation has occurred.
*****
root@rbknim:/export/fixes>
```

3. When Live Update is working as expected, run it without a preview, as shown in Example 8-38.

Example 8-38 Running without a preview

```
# nim -o cust -a lpp_source=aix_7_3_1_lpps -a fixes=update_all -a live_update=yes
-a live_update_data=liveupdate_data_mod1 mod1
```

The tail end of the completed update output should look like Example 8-39.

Example 8-39 Tail of output from completed Live Update

Requesting resources required for live update.

```
.....
```

Notifying applications of an impending live update.

```
.....
```

Creating rootvg for boot of surrogate.

```
.....
```

Starting the surrogate LPAR.

```
.....
```

Creating a mirror of the original LPAR's rootvg.

```
.....
```

Moving workload to surrogate LPAR.

```
.....
```

Blackout Time started.

```
.....
```

Blackout Time end.

```
.....
```

The workload is running on surrogate LPAR.

```
.....
```

Shutting down the Original LPAR.

```
.....
```

The live update operation succeeded.

File /etc/inittab has been modified.

File /etc/rc has been modified.

File /etc/services has been modified.

File /etc/vfs has been modified.

File /sbin/rc.boot has been modified.

One or more of the files that are listed in /etc/check_config.files changed.

See /var/adm/ras/config.diff for details.

The Live Update process uses a DLPAR to create an LPAR, so when you log in again, the ssh key has changed.

For more information about the Live Update process, see the following resources:

- ▶ [AIX Live Update Cookbook](#)
- ▶ [Live Update](#)



IBM i

Modernizing the IBM i environment involves more than updating user interfaces (UIs). It requires evolving existing applications and infrastructure to meet current business needs and use modern technologies. This process includes enhancing application architecture, optimizing data management, and integrating with technologies such as web services and mobile applications. A primary driver of IBM i modernization is the need to preserve the platform's reliable core while expanding its capabilities to support digital transformation.

A significant aspect of IBM i modernization is transforming code, such as RPG, into more contemporary and maintainable formats. This process often includes migrating to free-form RPG to improve readability and compatibility with other programming languages. Database modernization is also essential. It focuses on standardizing data access through SQL and improving database performance. In addition, modernization efforts typically involve redesigning UIs to create web-based or mobile-friendly front ends that offer a more intuitive and accessible user experience. The goal is to preserve the critical business logic and data within IBM i systems while enabling seamless integration into modern IT ecosystems.

The following topics are described in this chapter:

- ▶ 9.1, “IBM i modernization” on page 280
- ▶ 9.2, “Modern development environments on IBM i” on page 282
- ▶ 9.3, “Db2 for i” on page 301
- ▶ 9.4, “Open-source databases on IBM i” on page 306
- ▶ 9.5, “Traditional programming languages” on page 323
- ▶ 9.6, “Open-source programming on IBM i” on page 325
- ▶ 9.7, “Integrating traditional and open-source languages” on page 327

9.1 IBM i modernization

The IBM i platform has long provided a reliable and robust foundation for critical business applications across many organizations. Known for its integrated architecture and strong performance, it continues to support essential operations. However, the evolving business and technology landscape requires these established systems to adapt to new demands.

IBM i modernization spans several key dimensions:

- ▶ *Application modernization* focuses on updating UIs, often migrating from traditional “green screens” to more intuitive GUIs. It also involves refactoring or redesigning existing code, such as converting RPG to free-form RPG. A critical component is integrating IBM i applications with newer systems through application programming interfaces (APIs) and adopting modern software development practices, including DevOps.
- ▶ *Infrastructure modernization* includes upgrading physical hardware, such as migrating to IBM Power10 servers, virtualizing the operating environment, adopting cloud or hybrid cloud deployment models, and implementing automation to improve operational efficiency.
- ▶ *Data modernization* centers on enhancing data management practices, strengthening security, and improving data accessibility. This effort involves integrating data across disparate systems to create a unified information landscape and using data for advanced analytics and business intelligence.

IBM i modernization can be approached incrementally, allowing organizations to advance in manageable stages rather than through a disruptive, all-encompassing overhaul.

Recognizing that modernization is a flexible and adaptable journey rather than a fixed destination is essential for successful implementation.

IBM i modernization is a comprehensive process that evolves existing applications to integrate with modern technologies and meet current business demands. This approach uses existing investments while introducing new tools and architectural strategies. Modernization enhances several critical areas, including improving customer interactions, increasing development team efficiency, strengthening system security, and boosting overall performance. It also involves adapting applications for cloud environments to provide greater flexibility and scalability.

A primary driver of IBM i modernization is the need to enhance user experiences and improve developer productivity. Organizations also aim to strengthen security, improve performance, and integrate IBM i applications with modern technologies and cloud infrastructures.

Modernization efforts focus on key areas such as upgrading UIs from traditional green screens to modern GUIs. Other goals include improving data accessibility, enabling deeper insights through analytics integration, and helping ensure that systems can scale to support future business growth. Addressing the skills gap by making systems more accessible to developers with contemporary skill sets is another significant motivator.

The skills gap within the IBM i ecosystem is another critical driver. As experienced RPG developers approach retirement, there is a growing need to attract new talent that is equipped with contemporary programming skills. Modernization efforts that incorporate modern languages and development practices can make the platform more appealing to younger generations of developers. Enhancing agility and innovation is also a key objective. Modernized IBM i systems enable businesses to respond more swiftly to changing market dynamics, integrate cutting-edge technologies, and foster the development of innovative solutions. Strengthening security and compliance is a nonnegotiable driver. Earlier systems can harbor security vulnerabilities, and modernization allows organizations to implement up-to-date security measures and help ensure adherence to evolving regulatory requirements.

IBM i modernization offers several key benefits. It enables organizations to use existing IT investments, strengthen security, and improve customer experiences through modern interfaces. Development teams gain productivity through access to better tools and environments. Modernization also increases agility, allowing faster development and deployment of new features. Enhancing UIs and system performance contributes to a better experience for both employees and customers.

9.1.1 Strategies and approaches to IBM i modernization

Organizations embarking on IBM i modernization can adopt a range of strategies that are tailored to specific needs and objectives. The journey typically begins with a thorough assessment and planning phase. This phase involves analyzing the existing application portfolio and infrastructure to identify modernization requirements and develop a strategic roadmap.

One common approach to IBM i modernization is UI modernization, also known as screen refacing. This strategy focuses on transforming traditional character-based green screen interfaces into modern, intuitive web-based or GUIs. For developers working directly on the platform, IBM Rational Developer for i (IBM RDi) provides an integrated development environment (IDE) that is built on the Eclipse framework. IBM RDi is designed for creating and maintaining applications on IBM i systems.

Code modernization is another key area of IBM i modernization and can involve several techniques. IBM i Modernization Engine for Lifecycle Integration (Merlin) serves as a dedicated application development and modernization environment for IBM i users. Running within Red Hat OpenShift containers, Merlin provides tools that guide software developers in modernizing IBM i applications. These tools support RPG conversion, streamline the DevOps pipeline, and enable cloud integration.

RPG conversion automates the transformation of fixed-form RPG code into the more modern and flexible free-form RPG. Enhancing traditional programming languages is described in 9.5, “Traditional programming languages” on page 323.

Some organizations choose *language migration*, which involves rewriting applications in modern programming languages such as Java, PHP, Python, or .NET. The usage of modern open-source languages is described in 9.6, “Open-source programming on IBM i” on page 325. Another strategy is *code refactoring*, which focuses on restructuring and optimizing existing code to improve performance and maintainability without changing its external function.

Database modernization is essential for unlocking the full potential of the IBM i system. This process often involves converting older data constructs to create a fully relational database, which improves performance and makes the database more accessible to developers who are familiar with modern database concepts. Db2 for IBM i continues to evolve to support new data requirements within the IBM i environment. These capabilities are described in 9.3, “Db2 for i” on page 301.

9.2 Modern development environments on IBM i

Modernizing application development on IBM i presents significant opportunities but also faces a major challenge: a growing skills shortage. As traditional RPG and COBOL developers retire, the pool of professionals that are proficient in these languages continues to shrink. Concurrently, modern languages such as Python and Java are gaining traction on the platform. Bridging the gap between these technologies and the unique IBM i environment is essential. The scarcity of developers with both skills poses a challenge for organizations seeking to modernize IBM i applications. This shortage might slow digital transformation efforts and hinder innovation.

IBM introduced new development environments for IBM i programmers to help clients maintain and modernize applications, reduce the training that is required for IBM i developers, and bridge the skills gap.

This section describes two of those options: Visual Studio Code (VS Code), and Merlin.

9.2.1 Visual Studio Code as a development tool for IBM i

You can use VS Code as a development tool. It is an IDE. Over time, IBM provided several development tools for IBM i. However, many of these tools are now outdated and either withdrawn or no longer supported. Here are some of those outdated tools:

- ▶ Report Layout Utility (RLU)
- ▶ Screen Design Aid (SDA)
- ▶ File Compare and Merge Utility (FCMU)
- ▶ Advanced Printer Function (APF)
- ▶ Character Generator Utility (CGU)
- ▶ Data File Utility (DFU)

For more information about the status of these products, see [Software withdrawal and support discontinuance: IBM i Modernization Engine for Lifecycle Integration \(Merlin\) 1.0.0, IBM Rational Developer for i 9.6, and other selected functions in the IBM i portfolio](#).

Introducing Visual Studio Code

VS Code was developed by Microsoft for Windows in 2015 as a lightweight version of its commercial Visual Studio product. It was released as open source. VS Code is a cross-platform, multi-language code editor that you can use for modern development projects on IBM i.

For more information about VS Code, see [Visual Studio Code](#).

You can find the VS Code GitHub repository at this [repository](#).

All examples and screen captures in this section were produced by using Microsoft Windows 11 running Windows Subsystem for Linux (WSL) with Ubuntu 24.04.5 LTS.

Installing Visual Studio Code

You can install VS Code on various operating systems (OSs). Supported platforms are listed on the [install page for Visual Studio Code](#). Microsoft also offers an Insider version of VS Code, which includes the latest features that are released on a daily cycle. If you are interested in this version, you can find more information at the [Visual Studio Code insider's page](#).

Note: Install VS Code locally on your PC or Mac. It is not installed on your IBM Power server.

The layout of VS Code is similar to that of any modern IDE, such as Eclipse, NetBeans, or other comparable tools.

Figure 9-1 shows the layout of VS Code. If you are familiar with RDi, which is based on Eclipse, the interface should look familiar to you.

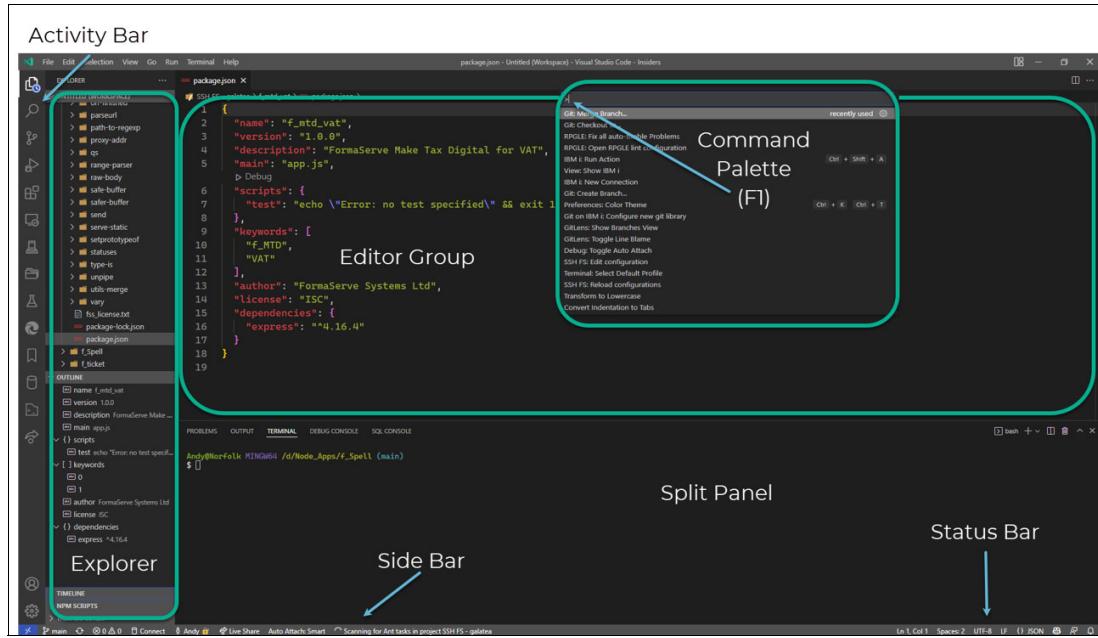


Figure 9-1 VS Code layout

When you first start VS Code, it presents a minimalist interface by design. This lean architecture means that the base installation includes a focused set of tools that are optimized for core source code editing and management in open-source programming languages. More functions are provided through an extensive ecosystem of extensions.

For example, if you need customized themes or specific key bindings, such as replicating a Notepad++ configuration, you can use the VS Code extension marketplace. This repository offers a wide range of extensions, including UI customizations, key map configurations, language-specific support, debugging tools, and integrated development features. With a catalog of more than 69,000 extensions, you have granular control over your VS Code environment, enabling a highly tailored and efficient workflow.

Figure 9-2 shows a subset of the catalog that is filtered by entering “IBM” in the search bar.

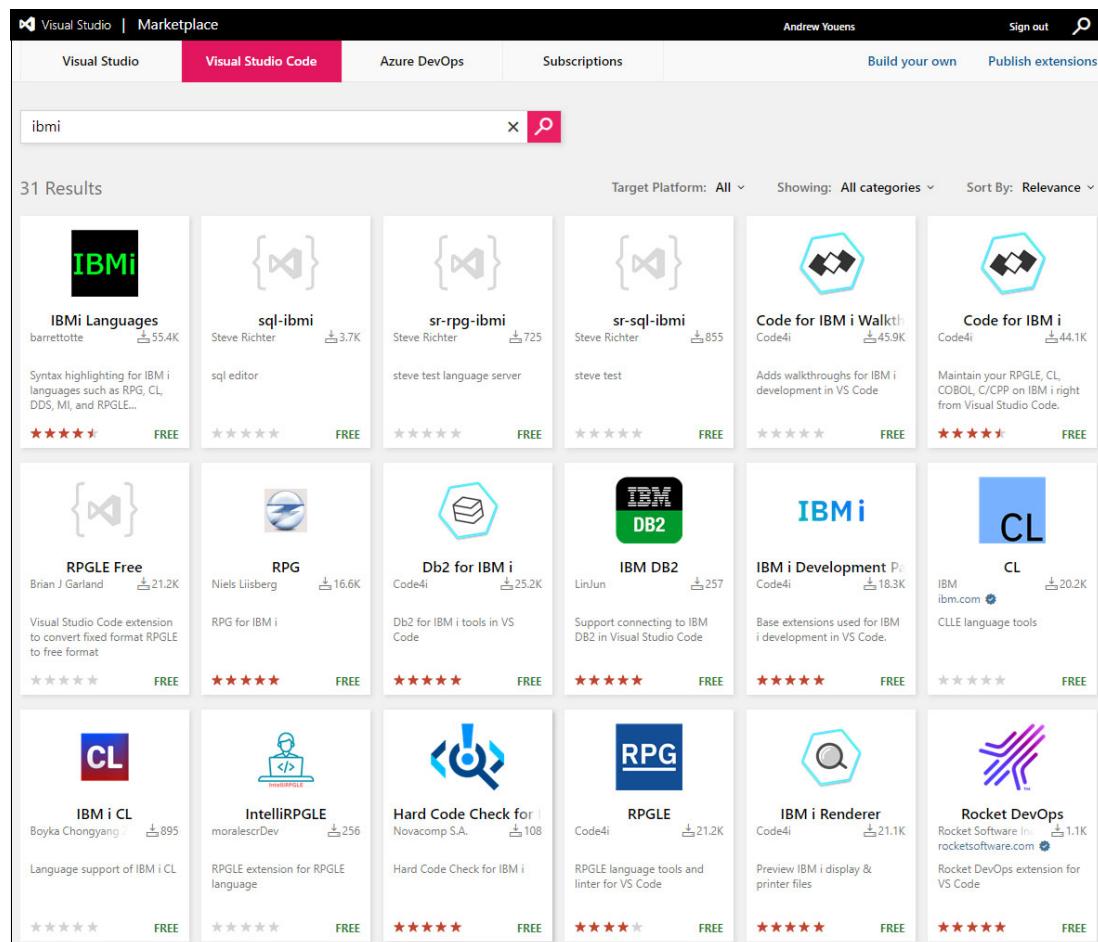


Figure 9-2 VS Code extensions

To begin IBM i development with VS Code, install the IBM i Development Pack extension. This extension provides the essential tools that are required to start working with IBM i in VS Code.

Figure 9-3 on page 285 shows the installation and layout of the IBM i Development Pack.

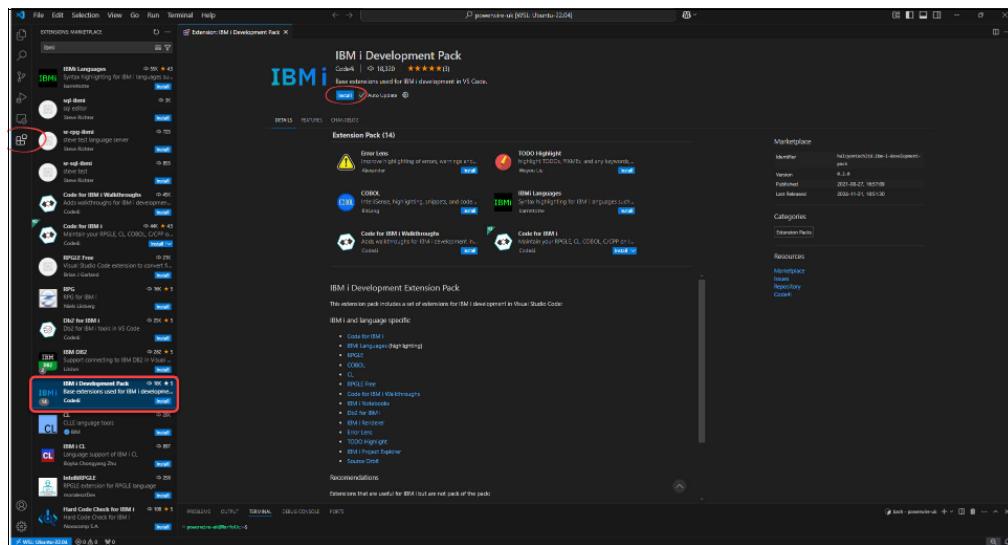


Figure 9-3 IBM i Development Pack

IBM i configuration for VS Code

After installing VS Code on your desktop, configure a connection to the IBM i system. This process establishes a Secure Shell (SSH) connection between your desktop and the IBM Power server.

Important: The SSH daemon must be running on your IBM i server for the connection to be successful.

To start the SSH service, run the Start TCP/IP Server (**STRTCPSSVR**) command, as shown in Figure 9-4. It is also possible to start this service by using IBM i Navigator.

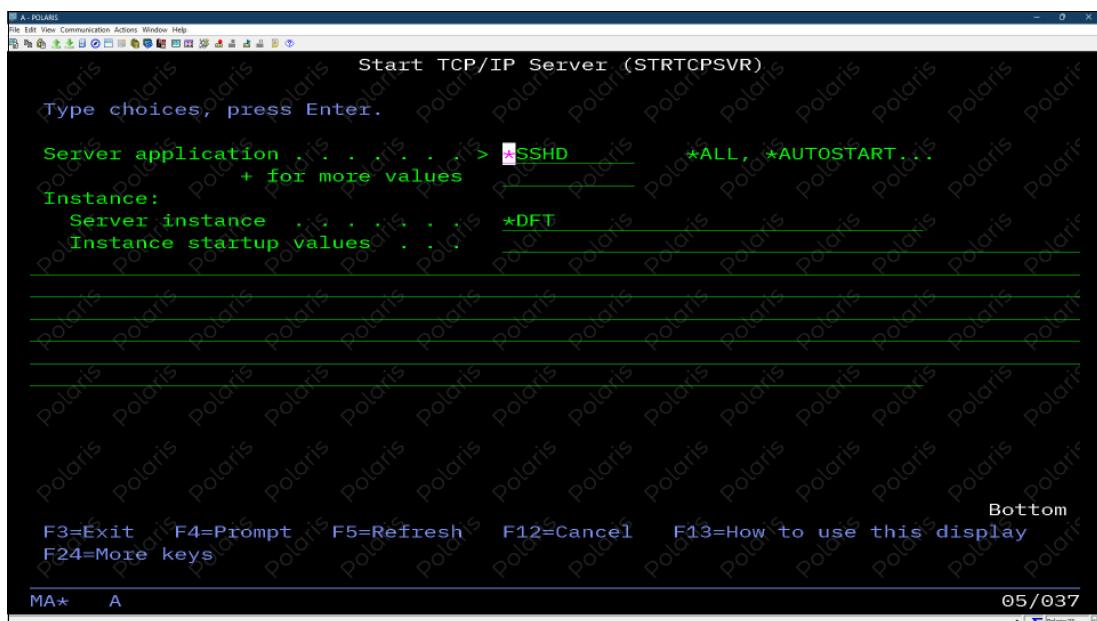


Figure 9-4 Starting the SSH service

When the extension is installed, you see a new icon in the activity bar of your VS Code instance, as shown in Figure 9-5.



Figure 9-5 Connection configuration tool

Selecting this icon opens the settings to configure the connection to the IBM i logical partition (LPAR). Enter the specific connection details in the configuration window, as shown in Figure 9-6.

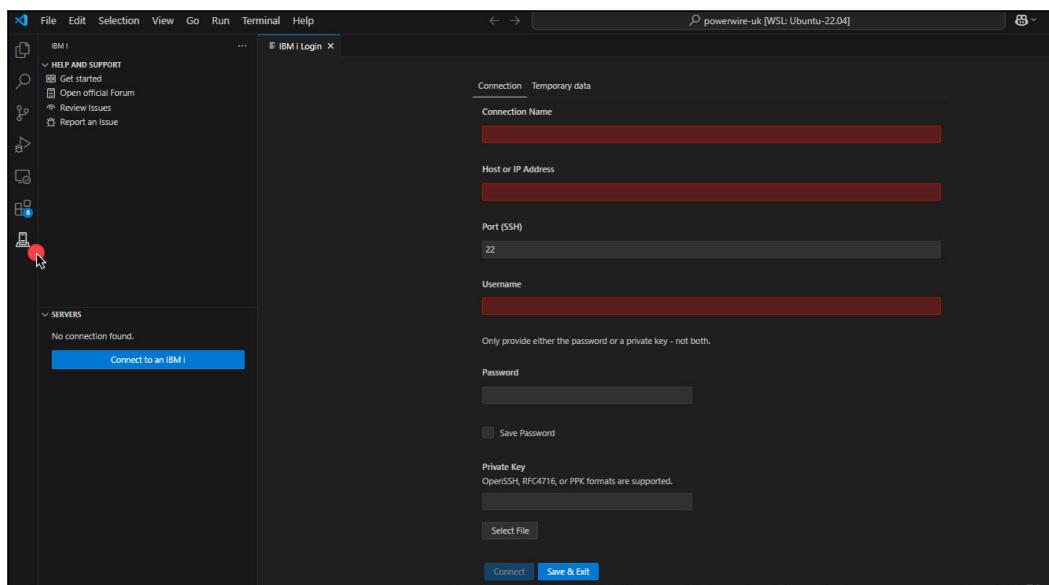


Figure 9-6 Entering details for the connection

When a connection is established, you can see your library list, objects, and Integrated File System (IFS) directories, as shown in Figure 9-7 on page 287.

The screenshot shows the Visual Studio Code interface connected to an IBM i LPAR named 'powerwire-uk'. The left sidebar displays the 'IBM i' Explorer view with several sections:

- HELP AND SUPPORT**
- USER LIBRARY LIST** (highlighted by a red arrow)
- OBJECT BROWSER** (highlighted by a red arrow)
 - Filter 1: POWERWIRE/*/* ('\$RCF')
 - qrpglesrc RPG Examples
 - inspec.rpgle H Spec Mixed Statements
 - lghf001.rpgle
 - new73_1.rpgle
 - new73_2.rpgle File Alias
 - new73_3.rpgle
 - new73_4.rpgle DataStructures
 - node01.rpgle Called From Nodejs
 - noxdb1.rpgle
 - noxdb2.rpgle
 - noxdb3.rpgle
 - proc1.rpgle PROCEDURES
 - utbrm00001.sqrpgle Old member UTBRM0...
 - vscode
- IFS BROWSER** (highlighted by a red arrow)
 - /home/andy
 - .ansible
 - .cache
 - .config
 - .mapepire
 - .node-red
 - .npm
 - .pm2
 - .sc
 - .ssh
 - .vscode
 - Anstible
 - curl
 - javasharedresources
 - ftp
 - python
 - rsync
 - source
 - .bash_history
 - .bashrc
 - .gitignore

The right panel shows a code editor window titled 'POWERWIRE/QRPGLSRC/NEW73_4.RPGLE' containing the following RPG code:

```

1  **Free
2
3
4
5
6
7
8
9
10 Dcl-F Camp Disk(*Ext) Usage(*Input : *Output : *Update) ;
11
12 Dcl-Ds dsCarp Extname('CARP') : *All ;
13 End-Ds;
14
15 // Use the data-structure dsCarp for all IO operations
16 Read Carr dsCarp;
17 Write Carr dsCarp;
18 Update Carr dsCarp;
19
20 Return ;

```

The bottom navigation bar includes tabs for PROBLEMS, OUTPUT, TERMINAL, DEBUG CONSOLE, PORTS, IBM I JOB LOG, and IBM I. The TERMINAL tab is selected, showing the command prompt: 'powerwire-uk@berfolk:~\$ []'

Figure 9-7 Visual Studio Code after connecting to the IBM i LPAR

Opening source code and compiling

To open source code for editing or browsing, locate the source in either the object browser or the IFS browser. Then, double-click the item to open it in the editor window.

When you are ready to compile the source code, right-click the item in the explorer. Select **Action** (Ctrl+E) to display the available options in the command window at the top of the window.

Figure 9-8 shows this process.

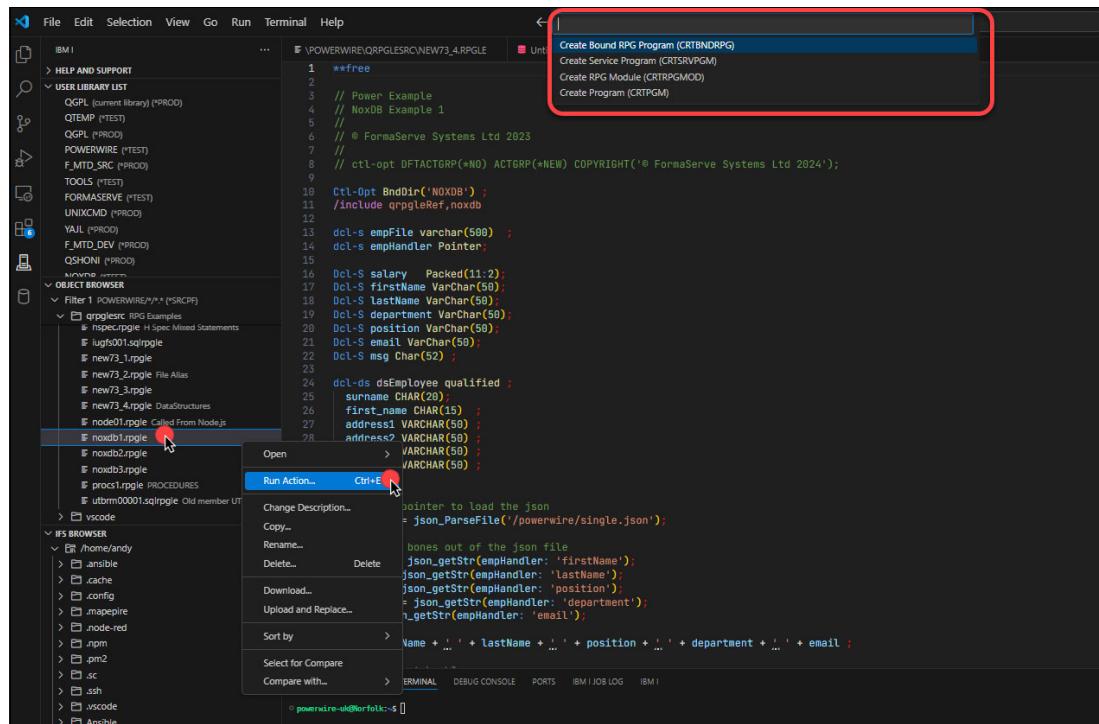


Figure 9-8 Selecting Action

Compile messages, including compile failure, appear in the tabbed terminal window, as shown in Figure 9-9.

```
* Executing task: IEM i: Create RPG Module (CRTRPGMOD)
Running Action: Create RPG Module (CRTRPGMOD) (11:23:02 AM)
Fetching errors for &OPENLIB/&OPENMBR.

* The terminal process terminated with exit code: 1.
* Terminal will be reused by tasks, press any key to close it.
```

Figure 9-9 Compile messages

Clicking an error shows the line where the compile failed.

Db2 for i extension

With the Db2 for i extension, you can run SQL statements to run any valid SQL command. You can install this extension through the [Visual Studio Code Marketplace](#).

Alternatively, you can install the extension by using the Extensions icon in the VS Code activity bar, as shown in Figure 9-10 on page 289. This extension is also included in the IBM i development pack and is automatically installed when you select that option.

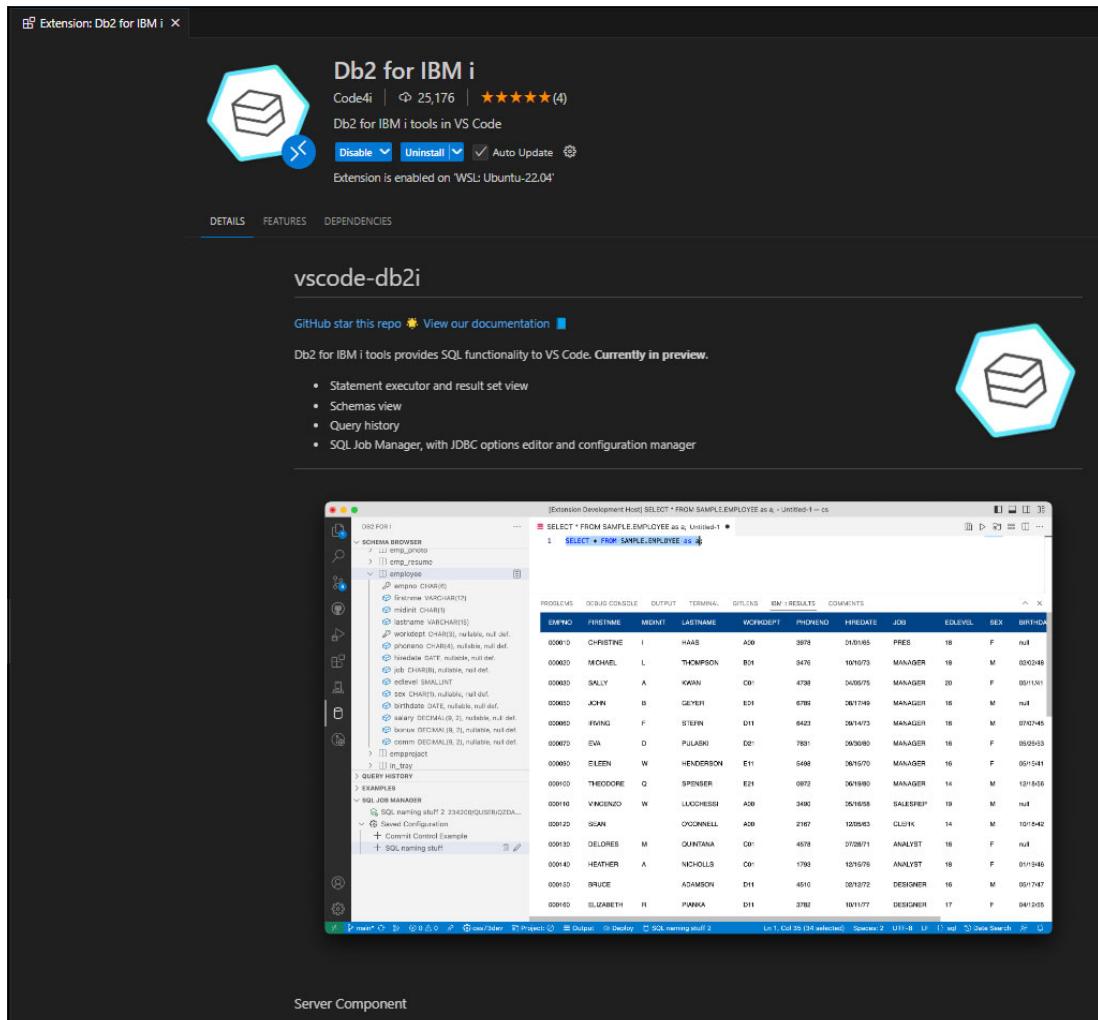


Figure 9-10 Db2 for IBM i extension

To run an SQL statement, open an existing .sql file or create one. The file opens in the editor window. When you are ready to run the SQL code, select **Run** in the upper right or use the keyboard shortcut Ctrl+R. Any result sets or errors are displayed in the **IBM i** tab.

Figure 9-11 shows this process.

```

File Edit Selection View Go Run Terminal Help
... Extensions (2) for IBM | Sensors X
Select *
  1 where true = true;
  2 /* Select employees between a date range, using from & to */
  3
  4 /* Select employees for System_Time From '2020-01-10-00:00:00' To '2020-01-10-17:00:00'
  5
  6 /* Get all employees between a date range */
  7
  8 /* From Employee For System_Time Between '2020-01-10-00:00:00' And '2020-01-10-17:00:00'
  9
 10 /* Stop monitoring - Don't need it anymore - taking up too much disk! */
 11
 12 Alter Table Employee
 13   Drop Versioning;
 14
 15 /* Let me update */
 16
 17 update select * from qdas.quoteoff;
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525
 526
 527
 528
 529
 530
 531
 532
 533
 534
 535
 536
 537
 538
 539
 540
 541
 542
 543
 544
 545
 546
 547
 548
 549
 550
 551
 552
 553
 554
 555
 556
 557
 558
 559
 560
 561
 562
 563
 564
 565
 566
 567
 568
 569
 570
 571
 572
 573
 574
 575
 576
 577
 578
 579
 580
 581
 582
 583
 584
 585
 586
 587
 588
 589
 589
 590
 591
 592
 593
 594
 595
 596
 597
 598
 599
 599
 600
 601
 602
 603
 604
 605
 606
 607
 608
 609
 609
 610
 611
 612
 613
 614
 615
 616
 617
 618
 619
 619
 620
 621
 622
 623
 624
 625
 626
 627
 628
 629
 629
 630
 631
 632
 633
 634
 635
 636
 637
 638
 639
 639
 640
 641
 642
 643
 644
 645
 646
 647
 648
 649
 649
 650
 651
 652
 653
 654
 655
 656
 657
 658
 659
 659
 660
 661
 662
 663
 664
 665
 666
 667
 668
 669
 669
 670
 671
 672
 673
 674
 675
 676
 677
 678
 679
 679
 680
 681
 682
 683
 684
 685
 686
 687
 688
 689
 689
 690
 691
 692
 693
 694
 695
 696
 697
 698
 699
 699
 700
 701
 702
 703
 704
 705
 706
 707
 708
 709
 709
 710
 711
 712
 713
 714
 715
 716
 717
 718
 719
 719
 720
 721
 722
 723
 724
 725
 726
 727
 728
 729
 729
 730
 731
 732
 733
 734
 735
 736
 737
 738
 739
 739
 740
 741
 742
 743
 744
 745
 746
 747
 748
 749
 749
 750
 751
 752
 753
 754
 755
 756
 757
 758
 759
 759
 760
 761
 762
 763
 764
 765
 766
 767
 768
 769
 769
 770
 771
 772
 773
 774
 775
 776
 777
 778
 779
 779
 780
 781
 782
 783
 784
 785
 786
 787
 788
 789
 789
 790
 791
 792
 793
 794
 795
 796
 797
 798
 799
 799
 800
 801
 802
 803
 804
 805
 806
 807
 808
 809
 809
 810
 811
 812
 813
 814
 815
 816
 817
 818
 819
 819
 820
 821
 822
 823
 824
 825
 826
 827
 828
 829
 829
 830
 831
 832
 833
 834
 835
 836
 837
 838
 839
 839
 840
 841
 842
 843
 844
 845
 846
 847
 848
 849
 849
 850
 851
 852
 853
 854
 855
 856
 857
 858
 859
 859
 860
 861
 862
 863
 864
 865
 866
 867
 868
 869
 869
 870
 871
 872
 873
 874
 875
 876
 877
 878
 879
 879
 880
 881
 882
 883
 884
 885
 886
 887
 888
 889
 889
 890
 891
 892
 893
 894
 895
 896
 897
 898
 899
 899
 900
 901
 902
 903
 904
 905
 906
 907
 908
 909
 909
 910
 911
 912
 913
 914
 915
 916
 917
 918
 919
 919
 920
 921
 922
 923
 924
 925
 926
 927
 928
 929
 929
 930
 931
 932
 933
 934
 935
 936
 937
 938
 939
 939
 940
 941
 942
 943
 944
 945
 946
 947
 948
 949
 949
 950
 951
 952
 953
 954
 955
 956
 957
 958
 959
 959
 960
 961
 962
 963
 964
 965
 966
 967
 968
 969
 969
 970
 971
 972
 973
 974
 975
 976
 977
 978
 979
 979
 980
 981
 982
 983
 984
 985
 986
 987
 988
 989
 989
 990
 991
 992
 993
 994
 995
 996
 997
 998
 999
 999
 1000
 1001
 1002
 1003
 1004
 1005
 1006
 1007
 1008
 1009
 1009
 1010
 1011
 1012
 1013
 1014
 1015
 1016
 1017
 1018
 1019
 1019
 1020
 1021
 1022
 1023
 1024
 1025
 1026
 1027
 1028
 1029
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038
 1039
 1039
 1040
 1041
 1042
 1043
 1044
 1045
 1046
 1047
 1048
 1049
 1049
 1050
 1051
 1052
 1053
 1054
 1055
 1056
 1057
 1058
 1059
 1059
 1060
 1061
 1062
 1063
 1064
 1065
 1066
 1067
 1068
 1069
 1069
 1070
 1071
 1072
 1073
 1074
 1075
 1076
 1077
 1078
 1079
 1079
 1080
 1081
 1082
 1083
 1084
 1085
 1086
 1087
 1088
 1089
 1089
 1090
 1091
 1092
 1093
 1094
 1095
 1096
 1097
 1098
 1099
 1099
 1100
 1101
 1102
 1103
 1104
 1105
 1106
 1107
 1108
 1109
 1109
 1110
 1111
 1112
 1113
 1114
 1115
 1116
 1117
 1118
 1119
 1119
 1120
 1121
 1122
 1123
 1124
 1125
 1126
 1127
 1128
 1129
 1129
 1130
 1131
 1132
 1133
 1134
 1135
 1136
 1137
 1138
 1139
 1139
 1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148
 1149
 1149
 1150
 1151
 1152
 1153
 1154
 1155
 1156
 1157
 1158
 1159
 1159
 1160
 1161
 1162
 1163
 1164
 1165
 1166
 1167
 1168
 1169
 1169
 1170
 1171
 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1179
 1179
 1180
 1181
 1182
 1183
 1184
 1185
 1186
 1187
 1188
 1189
 1189
 1190
 1191
 1192
 1193
 1194
 1195
 1196
 1197
 1198
 1199
 1199
 1200
 1201
 1202
 1203
 1204
 1205
 1206
 1207
 1208
 1209
 1209
 1210
 1211
 1212
 1213
 1214
 1215
 1216
 1217
 1218
 1219
 1219
 1220
 1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1239
 1240
 1241
 1242
 1243
 1244
 1245
 1246
 1247
 1248
 1249
 1249
 1250
 1251
 1252
 1253
 1254
 1255
 1256
 1257
 1258
 1259
 1259
 1260
 1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268
 1269
 1269
 1270
 1271
 1272
 1273
 1274
 1275
 1276
 1277
 1278
 1279
 1279
 1280
 1281
 1282
 1283
 1284
 1285
 1286
 1287
 1288
 1289
 1289
 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
 1299
 1300
 1301
 1302
 1303
 1304
 1305
 1306
 1307
 1308
 1309
 1309
 1310
 1311
 1312
 1313
 1314
 1315
 1316
 1317
 1318
 1319
 1319
 1320
 1321
 1322
 1323
 1324
 1325
 1326
 1327
 1328
 1329
 1329
 1330
 1331
 1332
 1333
 1334
 1335
 1336
 1337
 1338
 1339
 1339
 1340
 1341
 1342
 1343
 1344
 1345
 1346
 1347
 1348
 1349
 1349
 1350
 1351
 1352
 1353
 1354
 1355
 1356
 1357
 1358
 1359
 1359
 1360
 1361
 1362
 1363
 1364
 1365
 1366
 1367
 1368
 1369
 1369
 1370
 1371
 1372
 1373
 1374
 1375
 1376
 1377
 1378
 1379
 1379
 1380
 1381
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1389
 1389
 1390
 1391
 1392
 1393
 1394
 1395
 1396
 1397
 1398
 1399
 1399
 1400
 1401
 1402
 1403
 1404
 1405
 1406
 1407
 1408
 1409
 1409
 1410
 1411
 1412
 1413
 1414
 1415
 1416
 1417
 1418
 1419
 1419
 1420
 1421
 1422
 1423
 1424
 1425
 1426
 1427
 1428
 1429
 1429
 1430
 1431
 1432
 1433
 1434
 1435
 1436
 1437
 1438
 1439
 1439
 1440
 1441
 1442
 1443
 1444
 1445
 1446
 1447
 1448
 1449
 1449
 1450
 1451
 1452
 1453
 1454
 1455
 1456
 1457
 1458
 1459
 1459
 1460
 1461
 1462
 1463
 1464
 1465
 1466
 1467
 1468
 1469
 1469
 1470
 1471
 1472
 1473
 1474
 1475
 1476
 1477
 1478
 1479
 1479
 1480
 1481
 1482
 1483
 1484
 1485
 1486
 1487
 1488
 1489
 1489
 1490
 1491
 1492
 1493
 1494
 1495
 1496
 1497
 1498
 1499
 1499
 1500
 1501
 1502
 1503
 1504
 1505
 1506
 1507
 1508
 1509
 1509
 1510
 1511
 1512
 1513
 1514
 1515
 1516
 1517
 1518
 1519
 1519
 1520
 1521
 1522
 1523
 1524
 1525
 1526
 1527
 1528
 1529
 1529
 1530
 1531
 1532
 1533
 1534
 1535
 1536
 1537
 1538
 1539
 1539
 1540
 1541
 1542
 1543
 1544
 1545
 1546
 1547
 1548
 1549
 1549
 1550
 1551
 1552
 1553
 1554
 1555
 1556
 1557
 1558
 1559
 1559
 1560
 1561
 1562
 1563
 1564
 1565
 1566
 1567
 1568
 1569
 1569
 1570
 1571
 1572
 1573
 1574
 1575
 1576
 1577
 1578
 1579
 1579
 1580
 1581
 1582
 1583
 1584
 1585
 1586
 1587
 1588
 1589
 1589
 1590
 1591
 1592
 1593
 1594
 1595
 1596
 1597
 1598
 1599
 1599
 1600
 1601
 1602
 1603
 1604
 1605
 1606
 1607
 1608
 1609
 1609
 1610
 1611
 1612
 1613
 1614
 1615
 1616
 1617
 1618
 1619
 1619
 1620
 1621
 1622
 1623
 1624
 1625
 1626
 1627
 1628
 1629
 1629
 1630
 1631
 1632
 1633
 1634
 1635
 1636
 1637
 1638
 1639
 1639
 1640
 1641
 1642
 1643
 1644
 1645
 1646
 1647
 1648
 1649
 1649
 1650
 1651
 1652
 1653
 1654
 1655
 1656
 1657
 1658
 1659
 1659
 1660
 1661
 1662
 1663
 1664
 1665
 1666
 1667
 1668
 1669
 1669
 1670
 1671
 1672
 1673
 1674
 1675
 1676
 1677
 1678
 1679
 1679
 1680
 1681
 1682
 1683
 1684
 1685
 1686
 1687
 1688
 1689
 1689
 1690
 1691
 1692
 1693
 1694
 1695
 1696
 1697
 1698
 1699
 1699
 1700
 1701
 1702
 1703
 1704
 1705
 1706
 1707
 1708
 1709
 1709
 1710
 1711
 1712
 1713
 1714
 1715
 1716
 1717
 1718
 1719
 1719
 1720
 1721
 1722
 1723
 1724
 1725
 1726
 1727
 1728
 1729
 1729
 1730
 1731
 1732
 1733
 1734
 1735
 1736
 1737
 1738
 1739
 1739
 1740
 1741
 1742
 1743
 1744
 1745
 1746
 1747
 1748
 1749
 1749
 1750
 1751
 1752
 1753
 1754
 1755
 1756
 1757
 1758
 1759
 1759
 1760
 1761
 1762
 1763
 1764
 1765
 1766
 1767
 1768
 1769
 1769
 1770
 1771
 1772
 1773
 1774
 1775
 1776
 1777
 1778
 1779
 1779
 1780
 1781
 1782
 1783
 1784
 1785
 1786
 1787
 1788
 1789
 1789
 1790
 1791
 1792
 1793
 1794
 1795
 1796
 1797
 1798
 1799
 1799
 1800
 1801
 1802
 1803
 1804
 1805
 1806
 1807
 1808
 1809
 1809
 1810
 1811
 1812
 1813
 1814
 1815
 1816
 1817
 1818
 1819
 1819
 1820
 1821
 1822
 1823
 1824
 1825
 1826
 1827
 1828
 1829
 1829
 1830
 1831
 1832
 1833
 1834
 1835
 1836
 1837
 1838
 1839
 1839
 1840
 1841
 1842
 1843
 1844
 1845
 1846
 1847
 1848
 1849
 1849
 1850
 1851
 1852
 1853
 1854
 1855
 1856
 1857
 1858
 1859
 1859
 1860
 1861
 1862
 1863
 1864
 1865
 1866
 1867
 1868
 1869
 1869
 1870
 1871
 1872
 1873
 1874
 1875
 1876
 1877
 1878
 1878
 1879
 1880
 1881
 1882
 1883
 1884
 1885
 1886
 1887
 1888
 1889
 1889
 1890
 1891
 1892
 1893
 1894
 1895
 1896
 1897
 1898
 1899
 1899
 1900
 1901
 1902
 1903
 1904
 1905
 1906
 1907
 1908
 190
```

To run a Command Language or Control Language (CL) command from within your SQL script, prefix the CL statement with **c1:**, as shown in the following line:

```
c1: crtlib mylib
```

After the SQL runs, the create library command runs.

9.2.2 Merlin

The IBM i platform remains a cornerstone for businesses worldwide. However, the application development landscape is rapidly evolving. To succeed in this dynamic environment, IBM i development must adopt modern tools and practices. One such tool is Merlin, which supports this modernization effort.

Merlin is a comprehensive solution that is designed to modernize the IBM i development experience. It aligns IBM i application development with industry standards by integrating tools such as Jenkins for continuous integration, Git for source control management, and a browser-based version of VS Code for a modern development environment. This integration streamlines the development process and promotes collaboration, automation, and efficiency.

In addition to streamlining development, Merlin provides key modernization capabilities. One critical feature is the automated conversion of fixed-form RPG to free-form RPG. (The different versions of RPG are described in 9.5.1, “RPG” on page 323.) This conversion helps modernize codebases, improving readability, maintainability, and developer productivity.

Merlin also includes in-depth application impact analysis, which enables you to understand the potential consequences of code changes before implementation. This approach reduces the risk of errors and simplifies the modernization process.

Figure 9-13 shows an overview of Merlin’s capabilities.

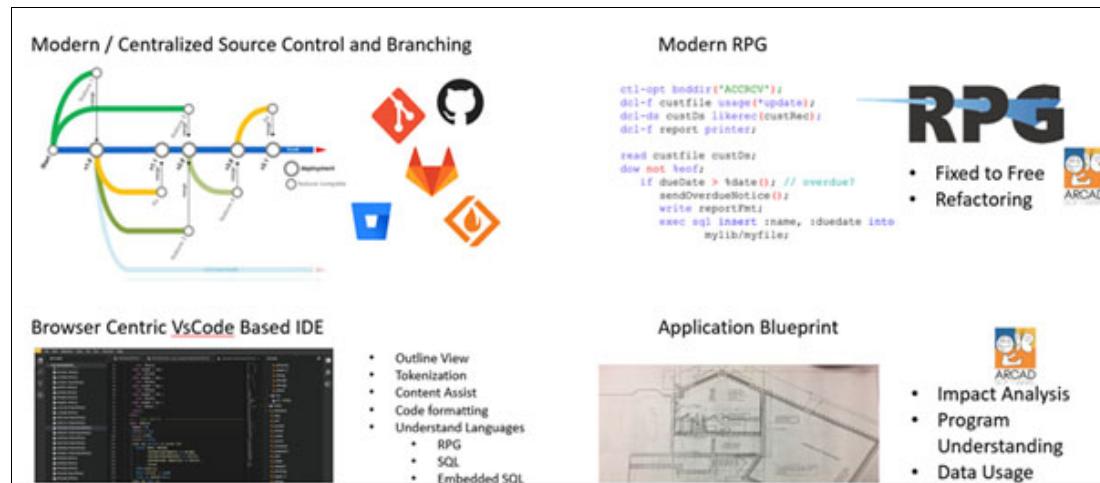


Figure 9-13 Merlin feature overview

Merlin benefits

For organizations invested in IBM i, Merlin does not replace existing tools such as RDi; it complements them. Developers now have options: Continue using the workstation-based IBM RDi or adopt the browser-accessible, container-based Merlin environment. Both approaches are fully supported. Merlin also offers more integrated capabilities through partnerships with solution providers such as ARCAD.

Merlin distinguishes itself by offering a fully integrated and supported suite of tools, including an IDE, plug-ins, and utilities that enable modern development practices. These features include seamless integration with continuous integration and continuous deployment or delivery (CI/CD) pipelines, native Git-based source control, and application impact analysis.

Merlin also incorporates DevOps expertise from ARCAD Software, supporting standard tools such as Git and Jenkins. It includes ARCAD functions that further enhance its modernization capabilities.

Merlin simplifies the provisioning of test environments for IBM i modernization by using IBM PowerVS and IBM PowerVC in an automated way. Instead of requiring deep technical expertise or manual setup, Merlin guides you through streamlined workflows that allow environments to be created quickly and consistently.

This capability is especially useful for teams that want to experiment or validate changes in a safe, repeatable setup without needing to manage infrastructure details. By reducing complexity behind the scenes, Merlin enables organizations to focus more on innovation and less on configuration.

Figure 9-14 provides a general overview of Merlin.

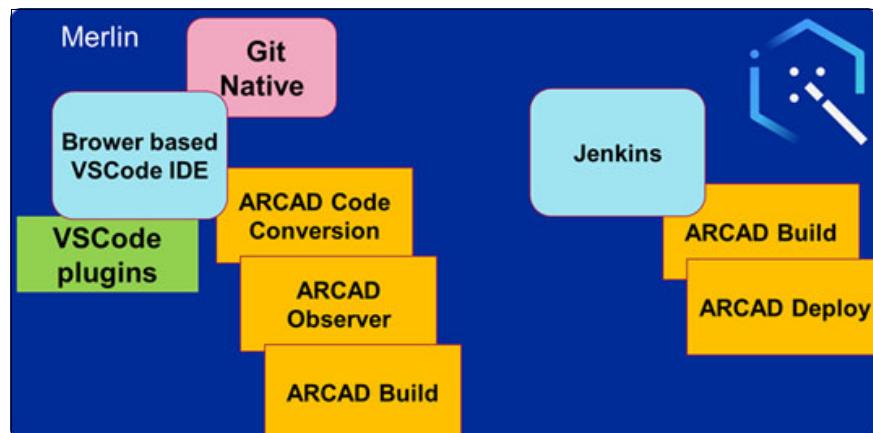


Figure 9-14 General overview of Merlin

Issues that Merlin helps solve in IBM i development

IBM i Merlin addresses several critical challenges that are faced by traditional IBM i development environments in today's rapidly evolving technology landscape. By introducing modern development practices to the IBM i platform, Merlin bridges the gap between its robust foundation and contemporary software development approaches.

Modern and centralized source control and branching

Traditional IBM i development often relies on library-based source management systems that lack the advanced versioning capabilities that are expected in modern development environments. Merlin addresses this limitation by natively integrating with Git to enable the following functions:

- ▶ Complete source code history with detailed tracking of changes, including who made them and when
- ▶ Support for concurrent development through branching and merging
- ▶ Isolation of feature development from production code

- ▶ Collaborative workflows that enable multiple developers to work on the same codebase without conflicts
- ▶ Simplified code reviews and approval processes before changes are deployed to production

This integration transforms IBM i development from an isolated, sequential workflow into a collaborative, parallel process. As a result, productivity and code quality are improved.

Modern RPG development

The migration from fixed-form to free-form RPG represents a modernization challenge for many IBM i environments. Merlin helps address this challenge through the following capabilities:

- ▶ Automated conversion tools that transform fixed-form RPG into modern free-form code
- ▶ Syntax highlighting and code completion that support the latest RPG syntax
- ▶ Refactoring tools that assist developers in adopting modern programming practices
- ▶ Consistency checks to help ensure that code adheres to current standards
- ▶ Support for the latest RPG features that are not available in fixed-form RPG

Merlin provides impact analysis that improves code comprehension and maintainability by visually mapping dependencies and potential change effects across the application. This modernization approach preserves existing business logic while making the code more maintainable and accessible to new developers who expect modern language features.

Browser-centric versus code-based IDE

Traditional IBM i development required installing and maintaining workstation software. Merlin's browser-based approach offers several advantages:

- ▶ Access to development environments from any device with a browser
- ▶ Consistent development experience regardless of location or hardware
- ▶ Reduced workstation management impact for IT departments
- ▶ Minimized security risks and centralized control by eliminating local installations
- ▶ Simplified onboarding for new developers through a familiar VS Code interface
- ▶ Compatibility with the VS Code extension ecosystem for customizing the development experience
- ▶ Real-time collaboration features that are not available in traditional desktop tools

Merlin's modern interface is designed to feel intuitive for existing IBM i developers, minimizing the need for extensive retraining. This accessibility transforms development from a location-dependent activity into a flexible process that can take place anywhere, at any time, with minimal setup.

Application blueprint and impact analysis

Understanding complex IBM i applications has traditionally been a challenge. Merlin addresses this challenge with application blueprint capabilities that provide the following features:

- ▶ Visual representation of application components and their relationships
- ▶ Impact analysis that identifies exactly what will be affected by code changes
- ▶ Dependency mapping to understand the full scope of modifications
- ▶ Risk assessment before changing critical systems
- ▶ Knowledge preservation by shifting understanding from tribal knowledge to documented structures

CI/CD integration

Traditional IBM i development often relied on manual processes to move code through development, testing, and production environments. These methods were time-consuming, error-prone, and difficult to standardize across teams. Merlin addresses this challenge by providing seamless integration with Jenkins and modern DevOps pipelines, enabling automated, consistent, and efficient deployment workflows.

When you use Merlin, your code changes can automatically trigger build processes, test suites, and deployment procedures. For example, when you commit code to a specific branch, Jenkins can automatically compile the RPG code, run unit tests, and prepare the objects for deployment. This automation reduces human error and helps ensure consistency in how code moves through development and production environments.

Merlin's CI/CD capabilities support practices such as continuous testing, in which automated test suites verify that new changes do not break existing functions. This process creates a safety net that enables developers to make changes with greater confidence, knowing that issues will be identified early in the development cycle rather than in production.

Figure 9-15 shows this approach.

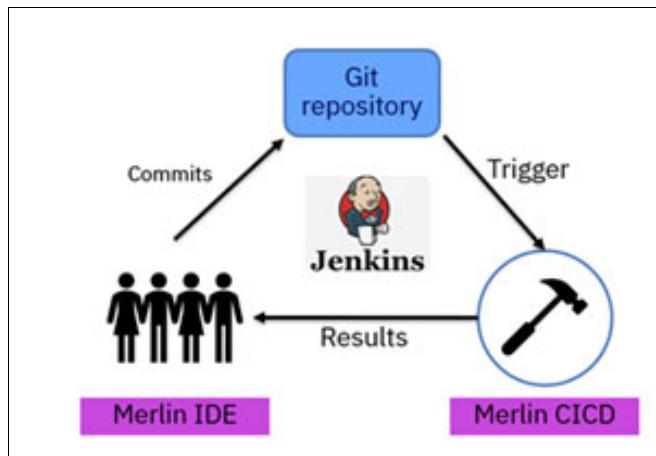


Figure 9-15 CI/CD for IBM

Environment provisioning

IBM i developers traditionally experienced delays when requesting new development or test environments. These environments often required manual setup by system administrators, which reduced productivity.

Merlin uses a container-based architecture to enable on-demand creation of development and test environments. Developers can provision environments in minutes to test new features or reproduce bugs. These environments are consistent and reproducible, eliminating the common “it works on my machine” issue.

The provisioning capabilities extend to creating isolated environments for specific development branches, enabling A/B testing scenarios, or supporting parallel development streams. This flexibility accelerates development cycles and improves software quality by allowing more thorough testing in environments that closely match production.

Cloud compatibility

As organizations increasingly adopt hybrid cloud strategies, IBM i applications sometimes are left behind due to their traditional infrastructure requirements. Merlin bridges this gap by supporting deployment in various environments, including on-premises Power servers, IBM Cloud (IBM Power Virtual Servers), or any cloud that supports Red Hat OpenShift.

This flexibility enables organizations to align IBM i development practices with broader cloud strategies. For example, a company can maintain production IBM i workloads on-premises while moving development and testing to the cloud to use elastic resources and reduce capital expenditure (CapEx).

Cloud compatibility enables global development teams to collaborate more effectively by accessing development environments from any location without complex virtual private network (VPN) configurations or remote desktop solutions. This capability is especially valuable for organizations with distributed teams or remote work policies.

Skills gap

The IBM i platform requires specialized knowledge that is becoming increasingly scarce as experienced developers retire. This shortage creates a skills gap that threatens the long-term sustainability of IBM i applications.

Merlin addresses this challenge by offering familiar, modern tools that appeal to developers with contemporary skills. The VS Code-compatible IDE enables developers who are experienced in web and open-source development to become productive with IBM i without extensive retraining.

Merlin's modernization tools transform code into maintainable, readable formats that align with current programming practices. This transformation helps new developers understand and modify existing applications, reducing the reliance on specialized knowledge that is held by a small group of experienced developers.

Development standardization

In many organizations, IBM i development practices evolved organically over time, resulting in inconsistent approaches across teams and projects. This lack of standardization complicates code maintenance, resource sharing, and onboarding of new developers.

Merlin provides a consistent, integrated tool set that promotes standardized development practices. It includes built-in code formatting, quality checks, and workflow templates to help teams align with best practices. Integration with Git supports standard branching strategies and code review processes that enhance overall software quality.

By centralizing development in a container-based solution, Merlin helps ensure that all developers use the same tools and configurations. This consistency eliminates the common “works on my machine” issue that is caused by inconsistent development environments.

Cross-platform development

Modern applications often span multiple platforms, with IBM i systems integrated with web, mobile, and cloud services. Traditional IBM i development tools focus solely on the platform, creating silos between development teams working on different components of the same application.

Merlin breaks down development silos by providing a cross-platform development experience. The VS Code-compatible IDE supports multiple languages and platforms, enabling developers to work on IBM i code alongside JavaScript, Python, and other languages that are used in connected systems.

This unified experience reduces context switching for developers working across platforms and promotes holistic application development. It enables teams to build end-to-end solutions that integrate IBM i transaction processing capabilities with modern user experience technologies, creating applications that use the strengths of both environments.

IBM i Merlin architecture and components

IBM i Merlin 2 delivers a modern development experience through an IBM Certified Container that is designed to operate seamlessly within Red Hat OpenShift. This containerized deployment streamlines setup, enhances scalability, and helps ensure consistent performance across diverse environments. Red Hat OpenShift supports deployment on-premises, on Power servers, and in cloud environments such as IBM Cloud (IBM Power Virtual Servers) and other compatible providers. For organizations already using cloud infrastructure, Merlin integrates efficiently into existing Red Hat OpenShift ecosystems. Merlin also supports Single Node Red Hat OpenShift, enabling a reduced footprint and simplified deployment on a single server, as described in “Single Node Red Hat OpenShift” on page 300.

The latest iteration of IBM i Merlin represents an advancement in IBM i development tools. It is designed as a complete, container-based solution that uses the capabilities of Red Hat OpenShift. By embracing cloud-native principles, Merlin delivers flexibility and scalability. Its microservices-based architecture provides specialized functions through standardized APIs, enabling modular and adaptable development.

As shown in Figure 9-16, Merlin's architecture is strategically designed around the separation of concerns. It isolates development activities, modernization functions (ARCAD tools), and DevOps processes (CI/CD) into distinct yet interconnected services. This design enables organizations to scale individual components based on specific requirements without affecting overall system stability or performance, optimizing resource usage and enhancing development agility.

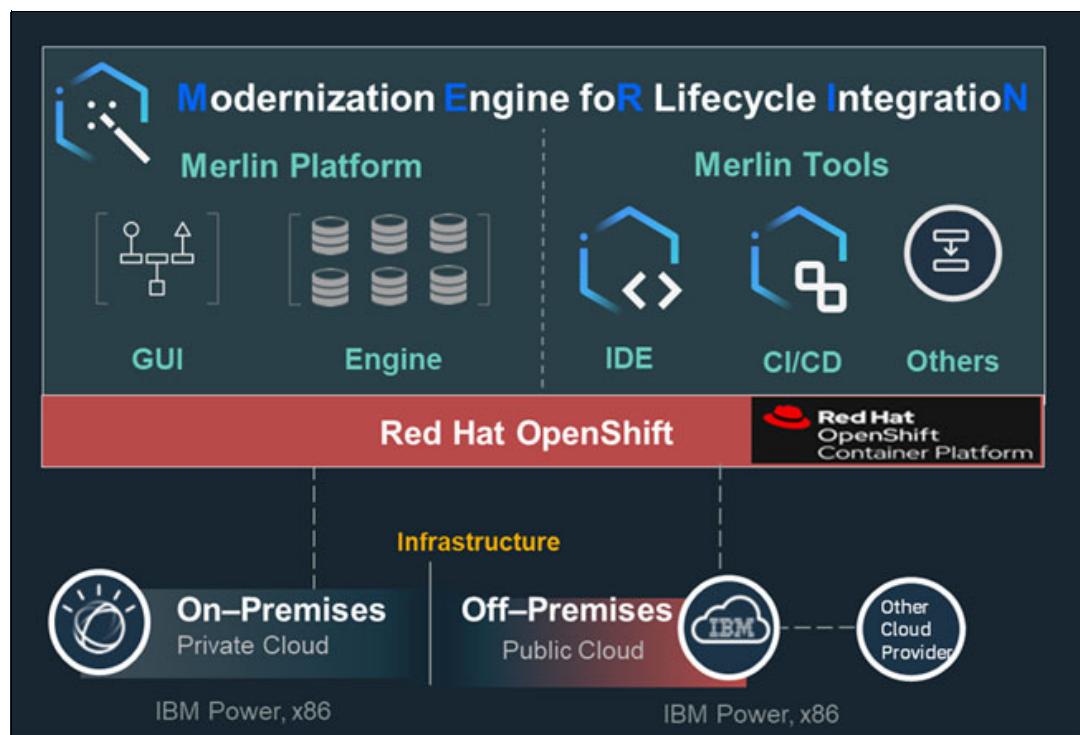


Figure 9-16 Merlin architecture

Red Hat OpenShift deployment framework

Merlin runs exclusively on Red Hat OpenShift, which provides the container orchestration foundation. This deployment model offers several advantages over traditional installations. The Red Hat OpenShift Container Platform supports automated scaling, self-healing capabilities, and consistent deployment across diverse infrastructure environments.

Merlin uses Red Hat OpenShift for deployment and lifecycle management, helping ensure optimal configuration across all infrastructure environments. This flexible deployment model allows Merlin to run on Power servers with Red Hat OpenShift installed directly, in IBM Cloud through PowerVS, or in any cloud environment that supports Red Hat OpenShift. This flexibility supports hybrid cloud strategies by enabling consistent development experiences across diverse deployment models.

Merlin is deployed as an IBM Certified Container, helping ensure compatibility and support across Red Hat OpenShift environments. This certification confirms that the container meets IBM standards for security, scalability, and integration.

DevSpaces integration

A key architectural component in Merlin 2 is its integration with Red Hat OpenShift DevSpaces (formerly CodeReady Workspaces). DevSpaces provides containerized development environments that are defined as code, which can be instantiated on demand.

Within Merlin, DevSpaces creates isolated development environments for each developer or project. These environments include all required tools, dependencies, and access configurations based on organizational standards. When a developer starts a session, DevSpaces provisions a container with the full development stack, accessible through a browser interface.

This architecture eliminates “works on my machine” issues by helping ensure all developers use identical environments. It simplifies onboarding by removing complex local setup requirements. New team members become productive within minutes because all tools and configurations are provisioned automatically.

IBM Licensing Service

Merlin integrates the IBM Licensing Service to provide a unified approach to license management across all components. This service offers visibility into license usage, monitors compliance, and simplifies the renewal process.

The licensing model follows a subscription-based approach that is aligned with modern cloud consumption patterns. Organizations pay based on usage rather than maximum capacity, optimizing costs while maintaining scalability. The service includes a dashboard that allows license administrators to monitor usage patterns and allocation efficiency.

IBM Certificate Manager

Security is a core aspect of Merlin's architecture, and is supported by the IBM Certificate Manager. This component manages the certificate lifecycle, including issuance, renewal, and revocation.

The Certificate Manager integrates with certificate authorities and supports development environments. It enforces certificate policies and rotation schedules to maintain security compliance.

By centralizing certificate management, Merlin reduces the risks that are associated with expired certificates and misconfigurations. The system includes monitoring features that alert administrators to upcoming expirations.

Browser-based IDE

The centerpiece of Merlin's development experience is its browser-based IDE, which is built on VS Code. The IDE provides code intelligence for IBM i languages, including RPG, CL, DDS, and SQL. It offers syntax checking, code completion, and inline documentation to accelerate the development process. The editor supports split views, enabling developers to view related files side by side, such as a display file and its associated RPG program.

Version 2 introduces enhanced debugging capabilities, including support for breakpoints, watch expressions, and call stack visualization. The IDE also includes integrated terminal access with IBM i command support, allowing developers to run commands directly within the development environment without switching contexts.

CI/CD toolchain

Merlin provides a CI/CD toolchain that is centered on Jenkins, with capabilities that are tailored for IBM i environments. This toolchain automates the software delivery process, from code commit to production deployment.

The build automation system supports both traditional IBM i build processes and modern approaches, such as IBM Integrated Language Environment (ILE) service programs. It integrates with source control systems to trigger builds automatically when code changes are detected, helping ensure that the latest code is consistently built and tested.

Merlin automates the deployment process by managing the movement of objects between IBM i environments. It maintains dependencies and helps ensure proper sequencing to reduce complexity and minimize risk during updates to critical systems.

The pipeline visualization tools provide real-time status updates and performance metrics, helping teams identify bottlenecks and improve delivery workflows. These visualizations include stage-level information, timing data, and failure analytics that help identify the root causes of pipeline issues.

ARCAD tools integration

Merlin 2 integrates with ARCAD tools, incorporating them as components within the solution. This integration delivers a seamless experience for code modernization, analysis, and build management.

Here is a list of the ARCAD tools:

► **ARCAD Transformer**

The Transformer component automates the conversion of fixed-form RPG code to modern free-form RPG. This tool goes beyond basic syntax transformation by applying contemporary programming patterns during the conversion process.

The implementation performs structural transformations that consider the overall organization of the code, including its syntax. For example, it recognizes common programming patterns and converts them into more maintainable modern structures. The tool also identifies potential issues, such as improper indicator usage or outdated file access methods.

The analysis capabilities enable developers to compare original and transformed code side by side, with explanations of each transformation. This transparency builds developer confidence in the automated process and serves as an educational resource for learning modern RPG techniques.

► ARCAD Observer

Observer provides application analysis and visualization capabilities that help developers understand complex IBM i applications. It identifies patterns and relationships within applications to support informed decision-making.

The application-mapping features generate visual representations of object dependencies, call hierarchies, and data flows. These maps help developers assess the impact of proposed changes before implementation, reducing the risk that is associated with modifying complex systems. The visualization tools enable developers to explore relationships interactively, focusing on areas of interest.

The code quality analysis evaluates applications against industry best practices and organizational standards. It identifies potential issues, such as unreachable code, inefficient algorithms, and security vulnerabilities. The analysis includes prioritized best practices for remediation based on potential impact.

Observer also generates comprehensive documentation that reflects the current state of applications. This feature is especially valuable for systems where original documentation might be outdated or missing. The generated documentation includes technical details for developers and business-oriented explanations for stakeholders.

► ARCAD Builder

Builder manages the build processes that are required for IBM i applications, helping ensure consistent and reproducible builds across environments. It includes support for build optimization to improve build times.

The dependency management features determine the correct build sequence based on object relationships, reducing reliance on manually maintained build scripts. The analysis considers both direct references and dependencies, such as data areas and shared files.

Incremental build capabilities optimize the process by rebuilding only the objects that are affected by changes. This approach reduces build times for large applications, where updates typically impact only a portion of the codebase.

Builder integrates with the CI/CD pipeline to provide build metrics and status updates. The implementation includes analytics that identify build patterns and suggest optimizations to improve efficiency.

Integration and workflow

Merlin distinguishes itself through the integration of its development components. The browser-based IDE connects to source control systems, triggering automated builds and deployments when code is committed. ARCAD tools provide insights directly within the development environment, enabling developers to make informed decisions without switching contexts.

Figure 9-17 shows how the development components fit together in a Merlin environment.

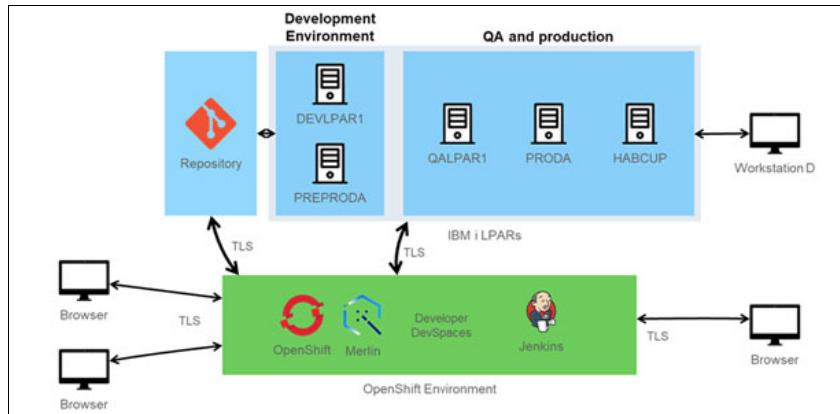


Figure 9-17 Merlin development topology

This integrated workflow supports the entire development lifecycle, from initial code creation through testing, deployment, and monitoring. The unified experience reduces context switching and tool fragmentation, enabling developers to focus on delivering business value rather than managing development infrastructure.

Single Node Red Hat OpenShift

The standard Red Hat OpenShift architecture, which is designed for high availability (HA), requires a minimum of three control plane nodes and three worker nodes. When deployed on an IBM Power server, this configuration becomes at least six LPARs, which can present a challenge for many clients.

To address this situation, Red Hat introduced Single Node Red Hat OpenShift, which is a streamlined deployment option that allows an entire Red Hat OpenShift cluster to run within a single LPAR on IBM Power. Given the inherent HA capabilities of IBM Power LPARs, the additional redundancy that is provided by a traditional multi-node Red Hat OpenShift cluster becomes less critical.

This approach is beneficial when deploying a Red Hat OpenShift cluster with an IBM i partition for a Merlin environment. Using Single Node Red Hat OpenShift in this scenario offers greater flexibility, optimizes resource usage, and simplifies deployment without compromising reliability within the robust IBM Power ecosystem.

Single Node Red Hat OpenShift is a specialized configuration of the Red Hat OpenShift Container Platform, and is designed for scenarios where a full-scale, HA cluster might be excessive. It consolidates control plane and worker node functions onto a single server, offering a streamlined architecture that is suitable for resource-constrained environments, edge computing deployments, and development or testing use cases.

Although Single Node Red Hat OpenShift does not provide the HA of multi-node clusters, it delivers core Red Hat OpenShift capabilities, such as container orchestration and application management, in a simplified and accessible format. This approach is valuable for remote locations with limited infrastructure or for developers seeking a rapid and efficient way to experiment with containerized applications.

Because this configuration runs on a single node, a failure of that node results in the failure of the entire Red Hat OpenShift instance. Therefore, Single Node Red Hat OpenShift is not recommended for production workloads that require continuous availability.

For more information about Merlin, including installation instructions and a user guide, see *Introducing IBM i Modernization Engine for Lifecycle Integration - Merlin*, SG24-8583.

9.3 Db2 for i

Db2 for IBM i is a relational database management system (RDBMS) that is deeply integrated with the IBM i operating system. This integration simplifies database management by eliminating the need for separate installation and configuration. As a native component of the OS, Db2 for IBM i offers a streamlined experience.

The platform supports a wide range of application needs, from traditional host-based systems to modern client/server and business intelligence solutions. Its comprehensive feature set includes triggers, stored procedures, and dynamic bitmapped indexing, enabling developers to build efficient and sophisticated database applications.

Db2 for IBM i adheres to industry-standard SQL, helping ensure compatibility and flexibility in data manipulation and querying. It seamlessly supports both IBM i files and SQL tables, enhancing its versatility. Tools such as IBM Db2 Query Manager and the SQL Development Kit for i provide developers with robust resources for interacting with and using the full capabilities of Db2 for IBM i.

9.3.1 Modernization techniques for IBM Db2 for i

Over time, Db2 for IBM i evolved from a traditional relational database into a comprehensive ecosystem that supports both structured and unstructured data, advanced analytics, and procedural programming. This section highlights key features of Db2 for IBM i that enhance SQL capabilities, enabling efficient data management, analysis, and performance optimization.

The complete code example for this section can be found at [this GitHub repository](#).

To view the catalog of services that are used in this section, run this SQL query:

```
select * from qsys2.services_info;
```

Building a sample schema

Before diving into advanced SQL techniques, you need a dataset to work with. Creating a sample schema in Db2 for i is straightforward. Run the following command:

```
call qsys.create_sql_sample ('SQLXXL');
```

With the schema set up, you can begin querying tables, such as EMPLOYEE and DEPARTMENT, to understand the available data.

Common Table Expressions

Common Table Expressions (CTEs) improve query readability and maintainability, especially when working with complex joins or recursive structures.

Example 9-1 demonstrates how to construct full employee names by using a CTE.

Example 9-1 Constructing full employee name

```
with emp_full_name as (
    select rtrim(firstname) || ' ' || midinit || ' ' || lastname as full_name, *
        from sqlxx1.employee
)
select * from emp_full_name
where full_name like '%JOHN%';
```

Recursive SQL

Recursive queries help with hierarchical data traversal, such as organizational structures.

Example 9-2 shows a query that retrieves department hierarchies. (Note the `connect_by` parameter.)

Example 9-2 Recursive query example

```
select level as dep_level,
       connect_by_root deptno as dep_root,
       sys_connect_by_path(trim(deptno), ' -> ') as dep_path,
       deptno, deptname, mgrno, admrdept
  from sqlxx1.department
 start with admrdept = 'A00'
 connect by nocycle prior deptno = admrdept
 order siblings by deptno;
```

OLAP and analytical processing

Db2 for i supports powerful OLAP functions for analytical queries.

Example 9-3 demonstrates cumulative sum calculations.

Example 9-3 Using cumulative sum calculations

```
select account_id, transaction_id, transaction_date, amount,
       sum(amount) over (partition by account_id order by transaction_date) as
       total
  from sqlxx1.account_transactions;
```

Additional OLAP functions such as `LAG()`, `LEAD()`, and `DENSE_RANK()` further enhance analytical capabilities.

Regular expressions

Db2 for i supports regular expressions for advanced text processing.

Example 9-4 shows the usage of `regexp_replace`.

Example 9-4 Regular expression support

```
values regexp_replace('AMOUNT: 123,456.78$', '[^0-9.]', ''');
```

This expression removes all non-numeric characters except for the decimal point.

Geospatial functions

Db2 for i includes geospatial functions for location-based data.

Example 9-5 adds latitude and longitude points to an organization table.

Example 9-5 Geospatial support

```
alter table sqlxx1.org add column location_point qsys2.st_point;
update sqlxx1.org
set location_point = qsys2.st_point('POINT(-74.1836 40.7261)')
where deptnumb = '10';
```

Distance calculations and mapping integrations are also possible by using built-in functions.

Compound statements

Db2 for i enables multiple SQL statements to be grouped by using compound statements, as shown in Example 9-6.

Example 9-6 Example of a compound statement

```
begin
  for
    select *
    from   qsys2.output_queue_entries_basic
    where  output_queue_name = 'QEJOBLOG'
    and    create_timestamp < now() - 7 days
  do
    call qcmandc('DLTSPLF FILE(' || SPOOLED_FILE_NAME
                 || ') JOB(' || JOB_NAME || ') SPLNBR(' || FILE_NUMBER || ')');
  end for;
end;
```

This example groups multiple statements into a structured execution block.

Stored procedures

Stored procedures enable reusable, parameterized SQL logic.

Example 9-7 shows a stored procedure for deleting a job log.

Example 9-7 Example of using stored procedures

```
create or replace procedure sqlxx1.delete_joblogs (
  in days_to_keep int default 7
)
  specific dltjoblogs -- This is the program name when we debug it
  language sql
  external action      -- let SQL know we are using the OS
  modifies sql data    -- This is not read-only
  set option dbgview = *source, output=*print, commit=*none, datfmt=*iso
begin
  for
    select *
    from   qsys2.output_queue_entries_basic outq
    where  output_queue_name = 'QEJOBLOG'
    and    create_timestamp < now() - days_to_keep days
  do
```

```
call qcmand('DLTSPLF FILE(' || SPOOLED_FILE_NAME
            || ') JOB(' || JOB_NAME || ') SPLNBR(' || FILE_NUMBER || ')');
end for;
end;
```

User-defined functions

User-defined functions (UDFs) allow custom SQL functions.

Example 9-8 shows a simple function that adds two integers.

Example 9-8 User-defined function example

```
create or replace function sqlxxl.add (
    x int,
    y int
)
returns int
begin
    return x + y;
end;
```

User-defined table functions

User-defined table functions return result sets.

Example 9-9 retrieves job logs older than a specified number of days.

Example 9-9 User-defined table function

```
create or replace function sqlxxl.joblogs_to_delete (
    older_than_days int default 7
)
returns table (
    spooled_file_name varchar(10),
    job_name varchar(28),
    file_number integer
)
specific joblog2dlt -- This is the program name when we debug it
language sql
no external action
reads sql data
set option dbgview = *source, output=*print, commit=*none, datfmt=*iso
begin
    return
        select spooled_file_name,
               job_name,
               file_number
        from qsys2.output_queue_entries_basic outq
        where output_queue_name = 'QEJOBLOG'
          and create_timestamp < now() - older_than_days days;
end;
```

JSON and XML processing

Db2 for i supports JSON and XML transformation.

Example 9-10 shows converting JSON to relational data by doing an HTTP request.

Example 9-10 JSON conversion

```
select * from json_table(
    qsys2.http_get('http://www.floatrates.com/daily/dkk.json'),
    'lax $.*'
    columns (
        code char(3) path '$.code',
        name varchar(32) path '$.name',
        rate float path '$.rate'
    )
);
```

Example 9-11 shows XML processing, also with an HTTP request.

Example 9-11 XML processing

```
select *
from xmltable(
    '/exchangerates/dailyrates/currency'
    passing xmpparse (
        document
        qsys2.http_get('https://www.nationalbanken.dk/api/currencyratesxml?lang=en')
    )
    columns
    currency_code char(3) path '@code',
    exchange_rate double path '@rate'
);
```

SQL triggers

Triggers automate database actions.

Example 9-12 shows an **INSTEAD OF** trigger that lets you update views.

Example 9-12 Example of an INSTEAD OF trigger

```
create or replace view sqlxx1.emp_full_name as (
    select empno,
        rtrim(firstname) concat ' ' concat midinit concat ' ' concat lastname as
    full_name
        from sqlxx1.employee
);
create or replace trigger sqlxx1.emp_full_name
instead of update on sqlxx1.emp_full_name
referencing new as new_row old as old_row
for each row mode db2row
begin
    if updating then
        update sqlxx1.employee
        set firstname = upper(regexp_substr( new_row.full_name, '^(\\w+)' )),
            lastname = upper(regexp_substr( new_row.full_name, '(\\w+$)' ))
        where empno = old_row.empno;
    end if;
end;
```

Temporal tables

Temporal tables track historical changes.

Example 9-13 is an example setup for a temporal table.

Example 9-13 Example setup of a temporal table

```
create table sqlxx1.department (
    deptno char(3) not null,
    deptname varchar(36) not null,
    start_ts timestamp(12) generated always as row begin,
    end_ts timestamp(12) generated always as row end,
    period system_time (start_ts, end_ts),
    primary key (deptno)
);
create or replace table sqlxx1.department_hist like sqlxx1.department;
alter table sqlxx1.department add versioning use history table
sqlxx1.department_hist;
select * from sqlxx1.department for system_time as of current timestamp - 6 months;
```

9.3.2 Summary

The advanced SQL features in Db2 for IBM i, including recursive queries, OLAP functions, geospatial processing, triggers, and temporal tables, reinforce its position as a leading-edge database platform. Mastery of these techniques enables the implementation of powerful data management and analytical workflows.

As Db2 for IBM i continues to evolve, proficiency in these advanced features becomes essential for building modern, reliable, and efficient database architectures.

9.3.3 References

Here is some additional documentation about the topics that are covered in this section.

- ▶ [All examples](#)
- ▶ [Cte](#)
- ▶ [OLAP](#)
- ▶ [Regex](#)
- ▶ [Geospatial](#)
- ▶ [Compound](#)
- ▶ [UDF](#)
- ▶ [UDTF](#)
- ▶ [JSON](#)
- ▶ [XML](#)
- ▶ [Triggers](#)

9.4 Open-source databases on IBM i

IBM i, known for its robust and secure environment, supports several open-source technologies to expand its database capabilities. This section introduces three open-source databases that can be effectively deployed on IBM i modernization projects: MariaDB, PostgreSQL, and SQLite.

MariaDB

MariaDB is a community-driven version of MySQL. MariaDB maintains strong compatibility while incorporating innovative features and performance enhancements.

Here are the key features of MariaDB:

- ▶ High performance
MariaDB is known for its speed and efficiency, making it suitable for demanding workloads.
- ▶ Robustness
MariaDB offers a stable and reliable platform.
- ▶ Active community
MariaDB benefits from a large and active community, which provides support, resources, and ongoing development.

MariaDB is ideal for a wide range of applications, including web applications, e-commerce platforms, and data warehousing.

PostgreSQL

PostgreSQL is a powerful, heavyweight, and feature-rich relational database system, which is known for its advanced features and strong emphasis on standards compliance.

Here are the key features of PostgreSQL:

- ▶ Extensibility
PostgreSQL supports a wide range of data types, including arrays, JSON, and spatial data.
- ▶ Advanced features
PostgreSQL offers features such as full-text search, triggers, and stored procedures.
- ▶ Active development
PostgreSQL is continuously evolving with new features and improvements.

PostgreSQL is suitable for complex applications, data analysis, and demanding workloads requiring advanced features.

SQLite

SQLite is a lightweight and embedded SQL database engine that is known for its simplicity. It is a great candidate for quickly and simply storing data.

Here are the key features of SQLite:

- ▶ Lightweight
SQLite is compact and simple to embed in applications.
- ▶ File-based
Stores data in a single file, making it simple to distribute and manage, especially when data replication is a requirement.
- ▶ No server required
Eliminates the need for a separate database server, which reduces complexity.

SQLite is ideal for mobile and embedded applications, small-scale projects, and situations where a lightweight database is required.

Choosing the database

This section provides a brief introduction to the open-source databases that are available for use on IBM i. Although Db2 for IBM i remains the primary database, open-source alternatives offer valuable options that should be considered when selecting a database for your application.

The best choice of database depends on your specific needs and requirements, and you must consider the following factors:

Workload	The type and volume of data that you are handling.
Features	The specific features that are required for your application (for example, full-text search and spatial data support).
Performance	The required performance and scalability of the database.
Simplicity	The level of technical expertise that is required to manage and maintain the database.

Table 9-1 provides a high-level comparison of the database options in several areas. Scores are 1-3, with 3 being the best.

Table 9-1 Open database comparison

Database	Workload	Features	Performance	Ease of Use
MariaDB	2	2	2	2
PostgreSQL	3	3	3	1
SQLite	1	1	1	3

9.4.1 MariaDB

MariaDB is an open-source database server that is developed by the original creators of MySQL. It serves as a drop-in replacement for MySQL and is recognized for its speed, scalability, and robustness.

MariaDB is a community-developed branch of MySQL, and many of its commands follow MySQL naming conventions. This familiarity makes it simple for developers migrating from MySQL to MariaDB. Its performance makes it a strong candidate for deployment on IBM i modernization projects.

Installing MariaDB

The simplest way to install MariaDB on IBM i is by using the IBM Access Client Solutions (ACS) open-source package management interface. Within ACS, find the two MariaDB packages and install them, as shown in Figure 9-18 on page 309.

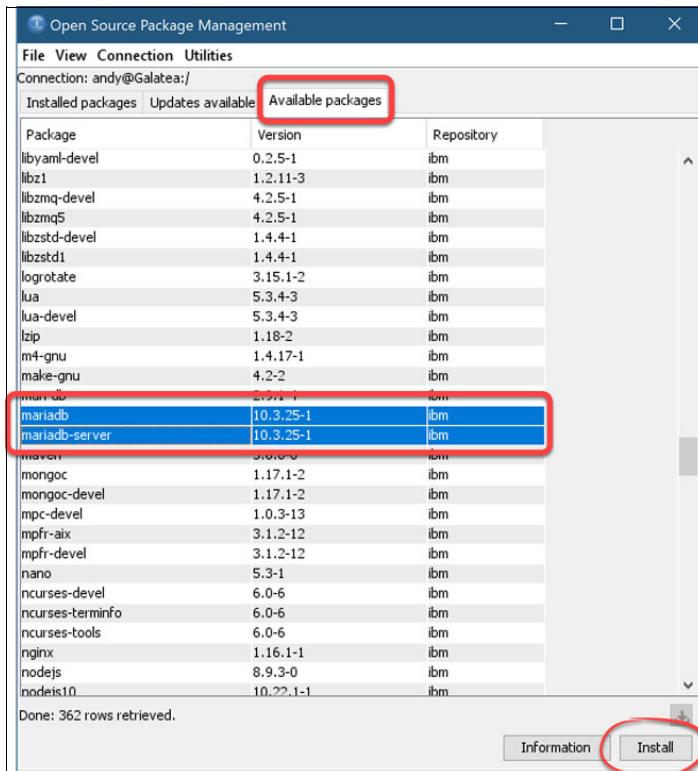


Figure 9-18 Package list

MariaDB can also be installed by using the YUM package manager. To install the required packages, open a shell session with QSEC0FR authority and run the following command:

```
yum install mariadb mariadb-server
```

Configuring MariaDB

Configure MariaDB by completing the following steps:

1. In a shell session, run the MariaDB setup procedure by running the following command:

```
mysql_install_db --user=mysql
```

2. Set the root password by running the following command:

```
mysqladmin -u root password *****
```

3. To apply the appropriate permissions, run the GRANT command. To do so, open a shell session and run the following commands:

```
mysql -u root -p
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'your password'
```

These commands assign privileges of type ALL to the root user when they connect from any IP address, as indicated by '%'. The *.* syntax specifies that the privileges apply to all databases and all tables. The IDENTIFIED BY clause secures the account with the specified password.

- Before starting the database server, help ensure that it is running on the correct IP address and port number by viewing the `my.cnf` file, which contains the configuration (Figure 9-19).

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "UNTITLED (WORKSPACE)". It includes "QOpenSys" (with an arrow), "etc" (with an arrow), and "mariadb" (with an arrow). Inside "mariadb", there are files like "init.d", "logrotate.d", "my.cnf" (highlighted with a red box), "mysql.cnf", "enable_encryption.preset", and "server.cnf".
- Editor:** The "my.cnf" file is open. The content is as follows:

```

1 #
2 # This group is read both by the client and the server
3 # use it for options that affect everything
4 #
5 [client-server]
6
7 #
8 # include *.cnf from the config directory
9 #
10 !includedir /QOpenSys/etc/mariadb/my.cnf.d
11
12 [mysqld]
13 bind-address = 0.0.0.0
14
15

```
- Terminal:** Shows the command "Andy@Mercury MINGW64 /c/Shared_Root/Web_Root/Lucky_Stars (main)" followed by a dollar sign.

Figure 9-19 MariaDB configuration

- Change the configuration to listen on all your IP addresses. By default, the file uses port 3306. If that port is in use by your corporate network, it can be changed here.

Starting the MariaDB server

To start the MariaDB server, run the following command from a shell session:

```
mysqld_safe --datadir=/QOpenSys/var/lib/mariadb/data
```

When this command finishes, it ties up the terminal, and if the terminal is closed, it stops the database server from running. This option is acceptable for initial testing.

After completing initial testing, it is a best practice to start the MariaDB server as a batch job. To do so, run the following command:

```
SBMJOB CMD(QSH CMD('/QOpenSys/pkgs/bin/mysqld_safe
--datadir=/QOpenSys/var/lib/mariadb/data')) JOB(STR_MARIA)
```

If you check the error log, which is shown in Figure 9-20 on page 311, you see that the server is ready and waiting for connections.

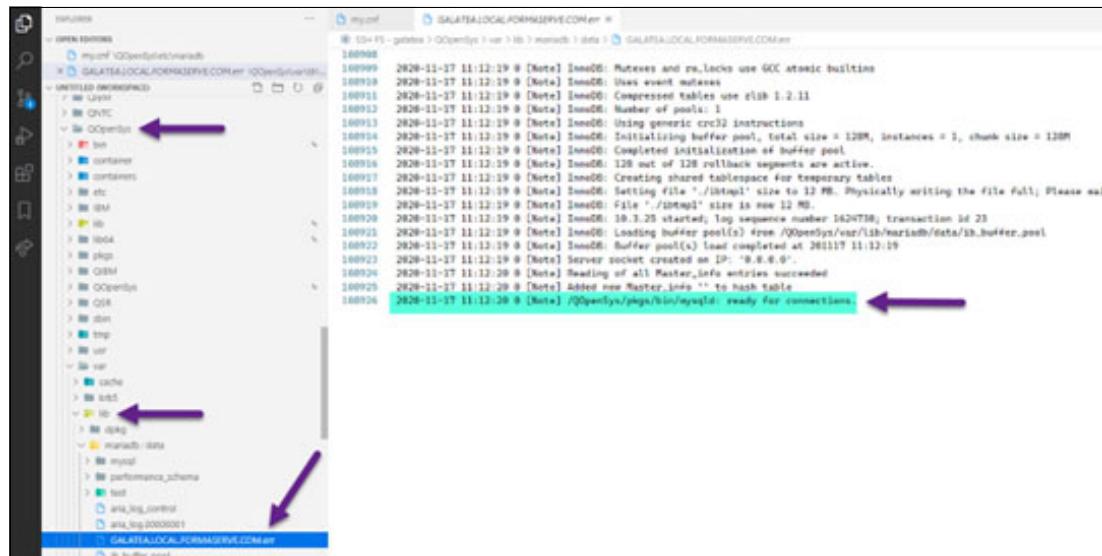


Figure 9-20 Validating that the server is ready

If you are using a 5250 terminal, you can run the **NETSTAT *CNN** command, and look for port 3306, as shown in Figure 9-21.

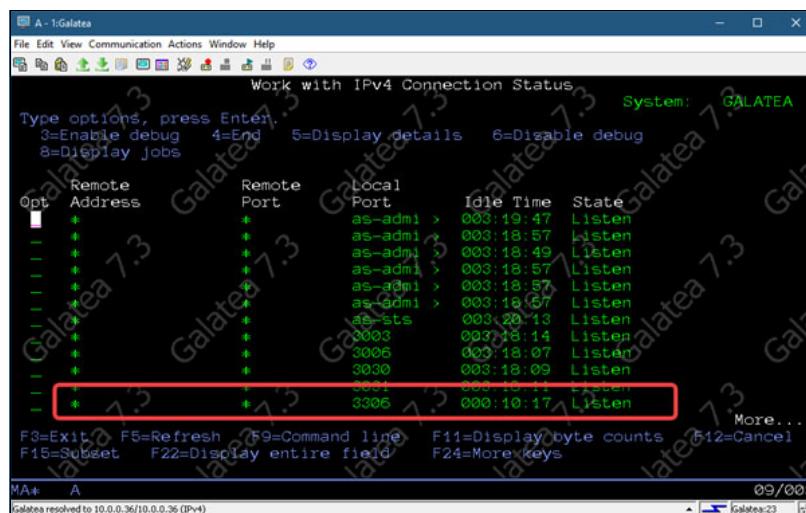


Figure 9-21 NETSTAT command

Alternatively, you can look for the job in WRKACTJOB. The server job runs by default in the QUSRWRK subsystem, as shown in Figure 9-22.

Work with Active Jobs						
CPU %:	Elapsed time:	Active jobs:	17/11/20			
Opt	Subsystem/Job	User	Type	CPU %	Function	Stat
	QUSRWRK	QSYS	SBS	.0	PGM-sshd	DEQ
	QPOZSPWT	QSECOFR	BCI	.0	PGM-sshd	SEL
	QPOZSPWT	QSECOFR	BCI	.0	PGM-sshd	SEL
	QPOZSPWT	ANDY	BCI	.0	PGM-bash	THD
	QPOZSPWT	QSECOFR	BCI	.0	PGM-sshd	SEL
	QPOZSPWT	ANDY	BCI	.0	PGM-sftp-server	SEL
	QPOZSPWT	QSECOFR	BCI	.0	PGM-sshd	SEL
	QPOZSPWT	ANDY	BCI	.0	PGM-bash	SEL
	QPOZSPWT	QSECOFR	BCI	.0	PGM-sshd	SEL
	QPOZSPWT	ANDY	BCI	.0	PGM-bash	SEL
	QPOZSPWT	QSECOFR	BCI	.0	PGM-sshd	SEL
	QPOZSPWT	ANDY	BCI	.0	PGM-bash	SEL
	QPOZSPWT	ANDY	BCI	.0	PGM-mysqld	SEL
	QSCWCHMS	QUSER	BCH	.0		DEQ
	QZDASOINIT	QSECOFR	PJ	.0		TIM
	QZRCSRVS	QSECOFR	PJ	.0		TIM

Figure 9-22 Using the WRKACTJOB screen

Stopping the server

If you want to stop the database server for any reason, run the following command in a shell session:

```
mysqladmin -u root -p shutdown
```

This command prompts you for the root password. After you provide the password, the job shuts down, as shown in Figure 9-23.

```
Running FormaServe Profile Script on Galatea  
GALATEA:/home/andy  
$ mysqladmin -u root -p shutdown ←  
Enter password: |
```

Figure 9-23 Stopping the database server

Starting MariaDB

To start MariaDb, complete the following steps:

1. Open a database client to access MariaDB. This example uses DBeaver, an open-source database client that you can download from [DBeaver](#).
2. When you open the DBeaver interface, create a database connection to your IBM i. Figure 9-24 on page 313 shows how to create this connection.

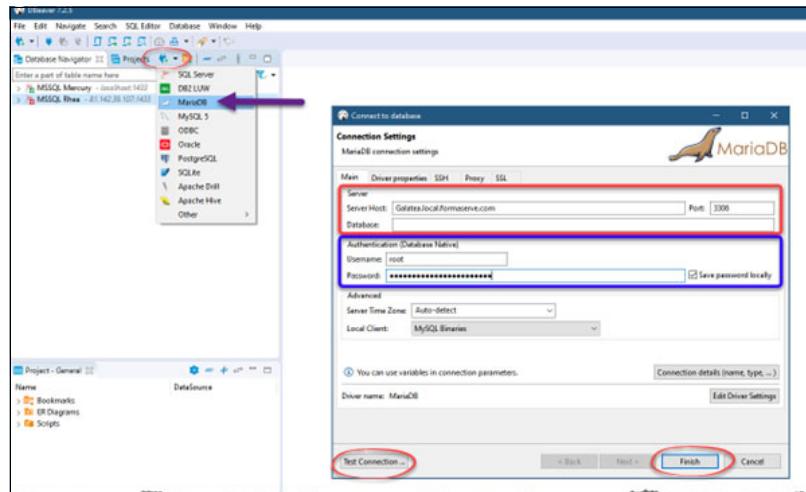


Figure 9-24 Creating the database connection

3. Click the New connection icon, select the MariaDB type, and complete the connection settings for both the server and user details.

Figure 9-25 show the MariaDB database on the IBM i.

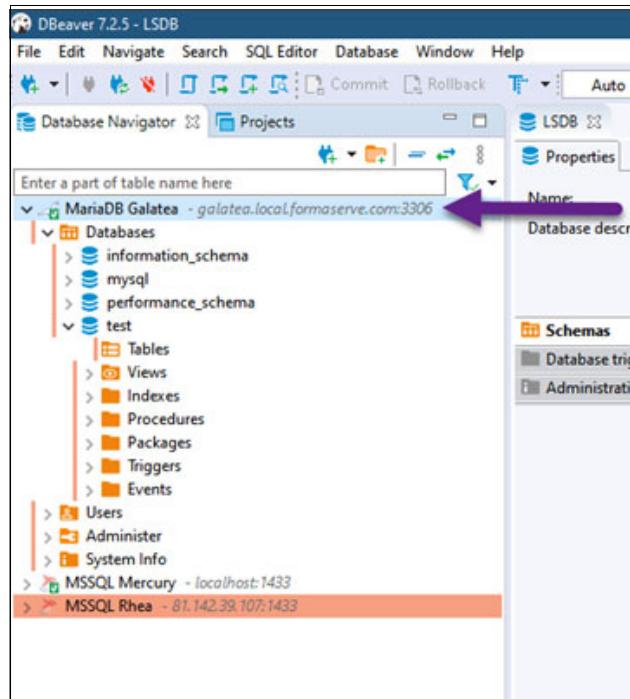


Figure 9-25 Seeing the connection

Creating a database and table

Create a database and create a couple of tables within it by downloading an SQL script from the [MariaDB site](#) that creates these database objects. When you have the script, complete the following steps:

1. Press **Ctrl+]** to open a new SQL window, and then press **Alt+X** to run the script, as shown in Figure 9-26.

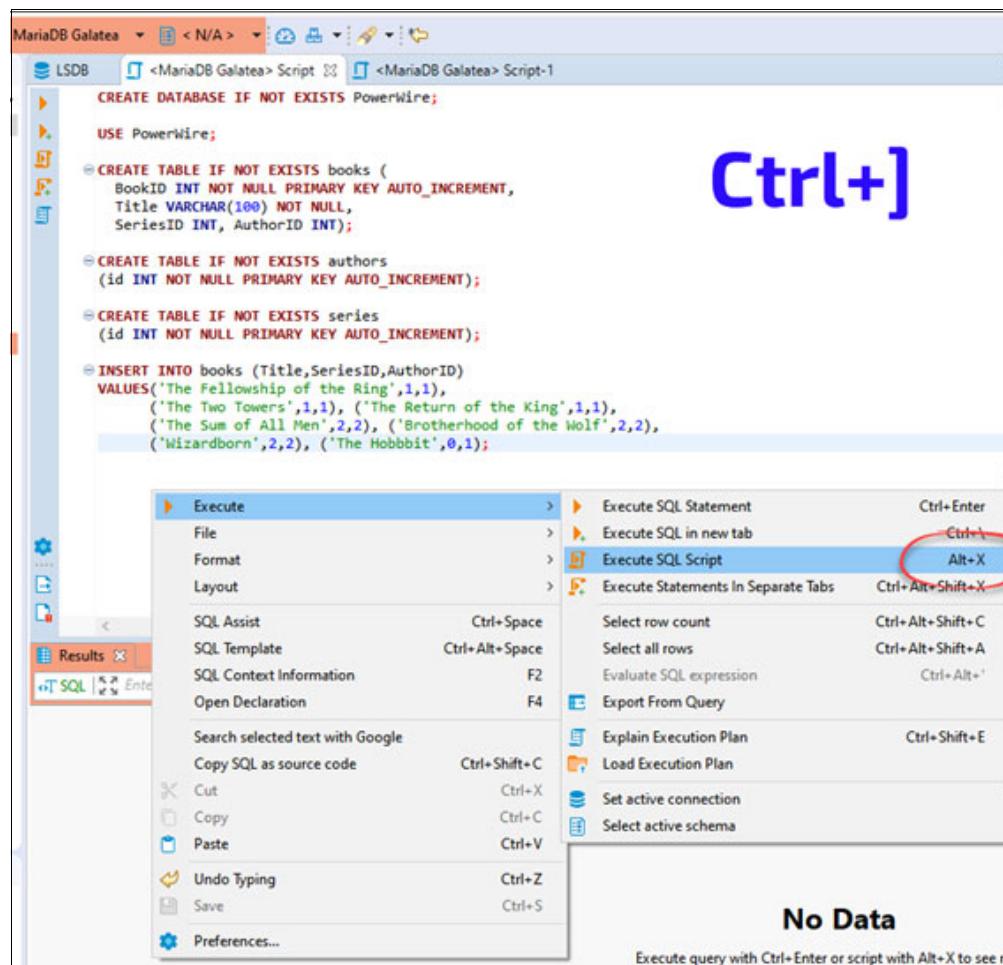


Figure 9-26 Running the script from an SQL window

2. Before the script runs, a warning appears and states that this server is a production server and prompts, “Do you confirm query execution?”, as shown in Figure 9-27 on page 315. Click **Yes To All**.

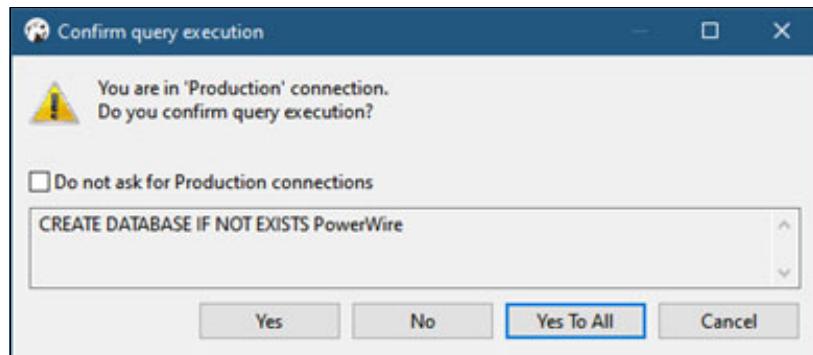


Figure 9-27 Confirm query execution

Figure 9-28 shows all the objects. You see that the new database and three tables exist.

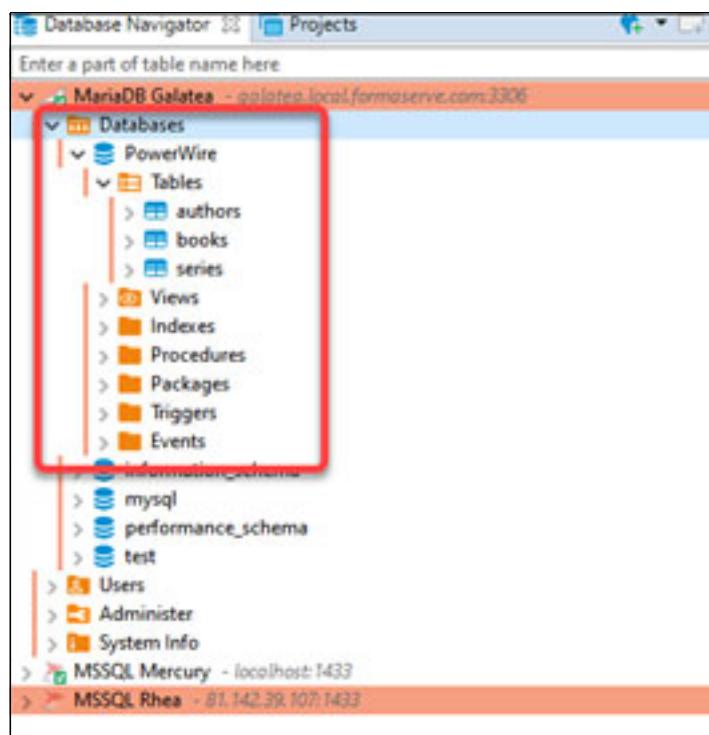


Figure 9-28 Database and table objects

3. To see whether the inserts worked, highlight the books table and press F4, as shown in Figure 9-29.

BookID	Title	SeriesID	AuthorID
1	The Fellowship of the Ring	1	1
2	The Two Towers	1	1
10	The Return of the King	1	1
11	The Sum of All Men	2	2
12	Brotherhood of the Wolf	2	2
13	Wizardborn	2	2
14	The Hobbit	0	1

Figure 9-29 Validating that the inserts worked

9.4.2 PostgreSQL on IBM i

This section explains how to install and use the PostgreSQL database on IBM i. PostgreSQL is an open-source RDBMS that is developed by a global community of contributors. It has been available since 1986 and is supported on the IBM i platform.

Installation

To install this database, use either IBM ACS and Open-Source Package Management, or YUM.

To use YUM, install PostgreSQL by running the following command in a shell:

```
yum install postgresql12-server postgresql12-contrib
```

Figure 9-30 shows the result.

```
Running FormaServe Profile Script on Galatea
andy@GALATEA ~ $ yum install postgresql12-server postgresql12-contrib
Setting up Install Process
Resolving Dependencies
--> Running transaction check
--> Package postgresql12-contrib.ppc64 0:12.2-2 will be installed
--> Processing Dependency: postgresql12-libpq = 12.2-2 for package: postgresql12-contrib-12.2-2.ppc64
--> Processing Dependency: lib:/QOpenSys/pkg/lib/postgresql12/lib/libpq.so.5(shr_64.o)(ppc64) for package: postgresql12-contrib-12.2-2.ppc64
--> Package postgresql12-server.ppc64 0:12.2-2 will be installed
--> Processing Dependency: postgresql12 = 12.2-2 for package: postgresql12-server-12.2-2.ppc64
--> Running transaction check
--> Package postgresql12.ppc64 0:12.2-2 will be installed
--> Package postgresql12-libpq.ppc64 0:12.2-2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Arch          Version       Repository      Size
=====
postgresql12-contrib   ppc64        12.2-2        ibm    5.9 M
postgresql12-server    ppc64        12.2-2        ibm    26 M
=====
Installing:
postgresql12           ppc64        12.2-2        ibm    2.7 M
postgresql12-libpq      ppc64        12.2-2        ibm    490 k

Transaction Summary
=====
Total download size: 35 M
Installed size: 131 M
Install          4 Packages
```

Figure 9-30 Installing with YUM

Figure 9-31 shows the completion message.



```
Installed:
  postgresql12-contrib.ppc64 0:12.2-2
                                postgresql12-server.ppc64 0:12.2-2

Dependency Installed:
  postgresql12.ppc64 0:12.2-2
                                postgresql12-libpq.ppc64 0:12.2-2

Complete!
andy@GALATEA ~ $
```

Figure 9-31 Completion message

Setup

To set up PostgreSQL on IBM i, complete the following steps:

1. Create a user who is called POSTGRES with the QSECOFR authority, and specify a home directory of /HOME/POSTGRES, by running the following commands:

```
CRTUSRPRF USRPRF(POSTGRES)
  USRCLS(*SECOFR)
  TEXT('Profile for PostgreSQL database')
  HOMEDIR('/home/postgres')
```

2. Create the home directory and ensure that the directories owner is correct by running the following commands in a shell:

```
mkdir /home/postgres
chown postgres /home/postgres
```

In a shell window, sign on by using the POSTGRES profile and run the following commands:

```
export PGDATA=/home/postgres
export PATH=$PATH:/QOpenSys/pkgs/bin/
```

These commands setup the path for all open-source packages and let Postgres know where its databases and environment are stored.

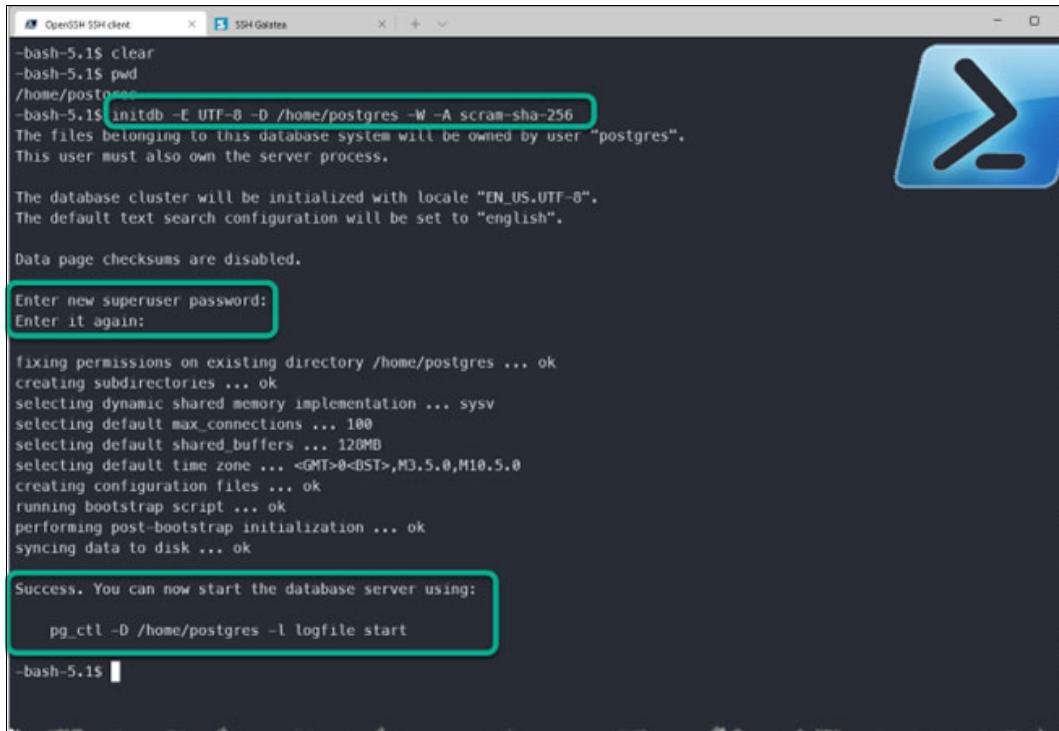
Important: Do *not* place any files in the POSTGRES home directory.

3. Initialize the configuration by running the following command:

```
initdb -E UTF-8 -D /home/postgres -W -A scram-sha-256
```

During the configuration, provide a POSTGRES profile for Postgres. It is not the same profile as the previously created POSTGRES user profile.

Figure 9-32 shows the database initialization.



The screenshot shows a terminal window titled "OpenSSH SSH client" with two tabs: "SSH Gates" and "SSH Gates". The main pane displays the following command and its output:

```
-bash-5.1$ clear
-bash-5.1$ pwd
/home/postgres
-bash-5.1$ initdb -E UTF-8 -D /home/postgres -W -A scram-sha-256
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "EN_US.UTF-8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

Enter new superuser password:
Enter it again:

fixing permissions on existing directory /home/postgres ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... sysv
selecting default max_connections ... 100
selecting default shared_buffers ... 120MB
selecting default time zone ... <GMT>0<BST>,M3.5.0,M10.5.0
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

Success. You can now start the database server using:

    pg_ctl -D /home/postgres -l logfile start

-bash-5.1$
```

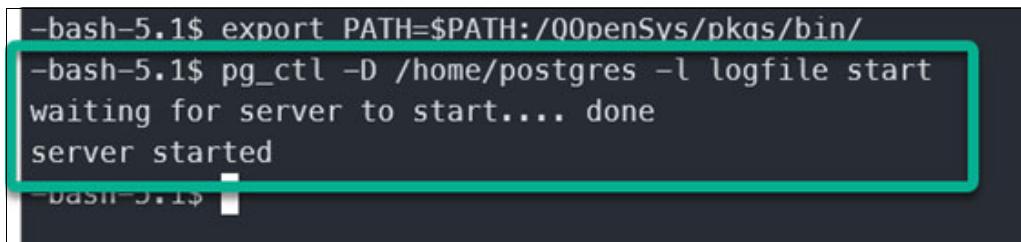
Figure 9-32 Database initialization

Starting the server

To start the server, run the following command:

```
pg_ctl -D /home/postgres -l logfile start
```

Figure 9-33 shows the results.

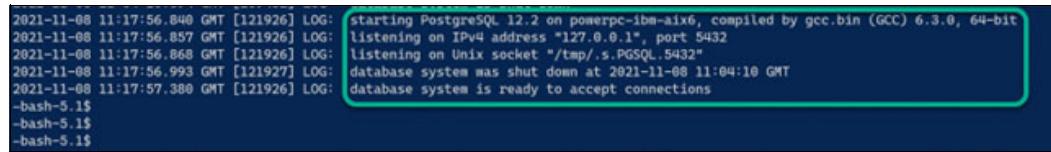


The screenshot shows a terminal window with the following command and its output:

```
-bash-5.1$ export PATH=$PATH:/QOpenSys/pkgs/bin/
-bash-5.1$ pg_ctl -D /home/postgres -l logfile start
waiting for server to start.... done
server started
-bash-5.1$
```

Figure 9-33 Starting the server

The Postgres server runs on port 5432. You can check the port by viewing the log file in the /home/postgres directory on your IFS, as shown in Figure 9-34.



The screenshot shows a terminal window with the following log entries:

```
2021-11-08 11:17:56.840 GMT [121926] LOG:  starting PostgreSQL 12.2 on powerpc-ibm-aix6, compiled by gcc.bin (GCC) 6.3.0, 64-bit
2021-11-08 11:17:56.857 GMT [121926] LOG:  listening on IPv4 address "127.0.0.1", port 5432
2021-11-08 11:17:56.868 GMT [121926] LOG:  listening on Unix socket "/tmp/.PGSQL.5432"
2021-11-08 11:17:56.993 GMT [121927] LOG:  database system was shut down at 2021-11-08 11:04:10 GMT
2021-11-08 11:17:57.380 GMT [121926] LOG:  database system is ready to accept connections
-bash-5.1$
-bash-5.1$
-bash-5.1$
```

Figure 9-34 Showing the PostGres server running on port 5432

To verify that PostgreSQL is accepting connections, you can use the **NETSTAT *CNN** command from a 5250 session, as shown in Figure 9-41 on page 339. This command helps confirm that port 5432 is active and listening for incoming connections.

Also, you can check whether the associated jobs are running in the QUSRWRK subsystem. These checks provide a quick way to validate that the PostgreSQL server is operational and ready to handle requests.

Work with Active Jobs							GALATEA
CPU %:	.3	Elapsed time:	00:18:23	Active jobs:	326		08/11/21 11:30:04 GMT
Opt	Subsystem/Job	User	Type	CPU %	Function	Status	
	QUSRWRK	QSYS	SBS	.0	PGM-sshd	DEQW	
	QP0ZSPWT	QSECOFR	BCI	.0	PGM-sshd	SELW	
	QP0ZSPWT	QSECOFR	BCI	.0	PGM-sshd	SELW	
	QP0ZSPWT	ANDY	BCI	.0	PGM-bash	SELW	
	QP0ZSPWT	QSECOFR	BCI	.0	PGM-sshd	SELW	
	QP0ZSPWT	POSTGRES	BCI	.0	PGM-bash	SELW	
	QP0ZSPWT	POSTGRES	BCI	.0	PGM-postgres	SELW	
	QP0ZSPWT	POSTGRES	BCI	.0	PGM-postgres	SELW	
	QP0ZSPWT	POSTGRES	BCI	.0	PGM-postgres	SELW	
	QP0ZSPWT	POSTGRES	BCI	.0	PGM-postgres	SELW	
	QP0ZSPWT	POSTGRES	BCI	.0	PGM-postgres	SELW	
	QP0ZSPWT	POSTGRES	BCI	.0	PGM-postgres	SELW	
	QP0ZSPWT	POSTGRES	BCI	.0	PGM-postgres	SELW	
	QSLICEV	QSYS	BCH	.0		DEQW	
	QSCWCHMS	QUSER	BCH	.0		DEQW	
====>							More...
Display instructions/keys							

Figure 9-35 Using the NETSTAT command to show that Postgres is running

Submitting to batch

To submit the start server job instead of running it, run the following command:

```
SBMJOB CMD(QSH CMD('/QOpenSys/pkgs/bin/pg_ctl -D /home/postgres -l logfile start')) JOB(STR_POSTGR)
```

Important: Ensure that you run this command by using the POSTGRES user profile.

Stopping the server

To stop the server, run the following command:

```
pg_ctl -D /home/postgres -l logfile stop
```

Using Postgres

To enter SQL statements into Postgres, use the **psql** command.

Figure 9-36 shows that the following SQL statements are performed:

1. Create a database.
2. Create a table.
3. Insert some records into the new table.
4. Check whether the records were inserted.

The screenshot shows a terminal window titled "SSH Gateway" with the title bar "OpenSSH SSH Client". The terminal displays the following PostgreSQL session:

```
Type "help" for help.
1
postgres=# Create database HRDATA ;
CREATE DATABASE
postgres=# CREATE TABLE employee (
    staff_id integer,
    first_name character varying(45) NOT NULL,
    last_name character varying(45) NOT NULL,
    first_name character varying(45) NOT NULL,
    last_name character varying(45) NOT NULL,
    address_id smallint NOT NULL,
    email character varying(50),
    username character varying(16) NOT NULL,
    last_update timestamp without time zone DEFAULT now() NOT NULL
);
CREATE TABLE
2
postgres=# insert into
    employee (staff_id, first_name, last_name, address_id, email, username)
values
    employee (staff_id, first_name, last_name, address_id, email, username)
postgres# values
3
postgres-#     (1, 'Andy', 'Youens', 331, 'Andy@formaserve.co.uk', 'Andy'),
postgres-#     (1, 'Jimmy', 'Greaves', 3231, 'Jim@formaserve.co.uk', 'Jimmy'),
postgres-#     (1, 'Dave', 'McKay', 765, 'Dave@formaserve.co.uk', 'Dave'),
postgres-#     (1, 'Micky', 'Hazard', 331, 'Micky@formaserve.co.uk', 'Micky'),
postgres-#     (1, 'Glen', 'Hoddle', 134, 'Glen@formaserve.co.uk', 'Glen');
INSERT 0 5
4
postgres# select * from employee ;
4
postgres-# +-----+-----+-----+-----+-----+-----+-----+
| staff_id | first_name | last_name | address_id | email           | username        | last_update    |
+-----+-----+-----+-----+-----+-----+-----+
|       1 | Andy      | Youens   |      331 | Andy@formaserve.co.uk | Andy            | 2021-11-08 13:24:54.407767
|       1 | Jimmy     | Greaves  |     3231 | Jim@formaserve.co.uk | Jimmy           | 2021-11-08 13:24:54.407767
|       1 | Dave      | McKay    |      765 | Dave@formaserve.co.uk | Dave            | 2021-11-08 13:24:54.407767
|       1 | Micky     | Hazard   |      331 | Micky@formaserve.co.uk | Micky           | 2021-11-08 13:24:54.407767
|       1 | Glen      | Hoddle   |      134 | Glen@formaserve.co.uk | Glen            | 2021-11-08 13:24:54.407767
(5 rows)
```

A red box highlights the final command and its output, which shows five rows of data inserted into the "employee" table.

Figure 9-36 Validating that commands ran

To quit the `psql` command, use `\q`, which takes you back to a Bash session.

9.4.3 SQLite

SQLite is a serverless, self-contained, and open-source database engine. Often referred to as an embedded database, SQLite runs as part of the application itself, eliminating the need for a central server or configuration tasks.

The primary advantage of SQLite is its ability to be embedded directly into programs, making it ideal for scenarios that require lightweight, temporary data storage. For applications with minimal infrastructure requirements or for short-term data persistence, SQLite offers a practical and efficient solution.

Installation

To install SQLite, you can use either IBM ACS, or YUM from an SSH shell session.

To install SQLite by using YUM, run the following command:

```
yum install sqlite3 sqlite3-devel
```

Using SQLite

To start SQLite, open an SSH shell session and enter SQLite3 and the name of a database that you want to work with. A new database is created if the file does not exist.

Now, complete the following steps:

1. Start SQLite and create an HRDATA schema.
2. Create the employee table.
3. Insert records into the employee table.
4. List all employee records.

Figure 9-37 shows the creation of a new table that is called employee in the HRDATA schema, and inserting a couple of records.

The screenshot shows a terminal window with the following sequence of SQLite commands:

1. The user connects to the SQLite database named "hrdata".
2. The user creates a table named "employee" with two columns: "first_name" and "surname".
3. The user inserts two records into the "employee" table.
4. The user runs a SELECT query to list all records from the "employee" table, which returns two rows: "Emily|Youens" and "Jessica|Jackman".

Figure 9-37 Using SQLite

You are now in the database command-line interface (CLI). Here, you can run any standard SQL statement, but each SQL statement must end in a semi-colon character.

Using .DOT commands

While you are in the SQLite3 CLI, there are numerous commands that you can run to show information about your database. These commands all start with the full stop character.

A summary of these commands is shown in Table 9-2. For the full list of these commands, see [Command Line Shell For SQLite](#).

Table 9-2 Commands for SQLite

Command	Description
.tables	List all tables in the database.
.databases	List all the databases.
.log	Switch logging on or off.
.quit	Exit SQLite command processing. (You can also use Ctrl-C.)
.help	To open the help page.

Figure 9-38 shows entering a couple of .DOT commands and the output that they produce:

- ▶ The **.databases** command shows all the databases.
- ▶ The **.tables** command shows all the tables in this database.
- ▶ The **ls -al** command shows the file that stores everything in the database. The complete database is in a single file (**hrdata**).

The screenshot shows a terminal window with three tabs labeled "Galatea". The current tab displays the following SQLite session and file listing:

```
andy@GALATEA ~/hr $ sqlite3 hrdata
SQLite version 3.16.2 2017-01-06 16:32:41
Enter ".help" for usage hints.
sqlite> .databases
main: /home/andy/hr/hrdata
sqlite> .tables
contacts employee
sqlite> l
andy@GALATEA ~/hr $ ls -al
total 184
drwxr-sr-x  2 andy    0          8192 Feb 14 15:01 .
drwxr-sr-x  26 andy   0          53248 Feb 14 14:42 ..
-rw-r--r--  1 andy    0         20480 Feb 14 15:01 hrdata
andy@GALATEA ~/hr $
```

Three specific lines are highlighted with yellow boxes and numbered 1, 2, and 3:

- 1 **main: /home/andy/hr/hrdata**
- 2 **contacts employee**
- 3 **-rw-r--r-- 1 andy 0 20480 Feb 14 15:01 hrdata**

Figure 9-38 Using commands in SQLite

9.4.4 Comparison

With multiple database options that are available on IBM i, evaluate which solution best meets the needs of your application environment. Sometimes, different areas of an application suite might benefit from using different databases.

Table 9-3 on page 323 provides a comparison between MariaDB and SQLite.

Table 9-3 Comparison between MariaDB and SQLite

Comparison	SQLite	MariaDB (MySQL)	Comments
Column types	Supports only the Blob, Integer, Null, Text, and Real data types.	Supports the full set of data types.	MariaDB is more flexible regarding data types.
Storage and portability	Stores the database in a single file, making it portable and small in size. No configuration is required.	With a bigger footprint, MariaDB is scalable and can handle a bigger database with less impact.	SQLite is suitable for smaller databases. Optimization is simpler with MariaDB.
Security	SQLite does not have a built-in authentication mechanism. The database files can be accessed by anyone.	MariaDB comes with integral security features, which include authentication.	If you have any security concerns and must restrict access, use MariaDB.

SQLite is ideal in the following situations:

- ▶ Developing small, stand-alone apps
- ▶ Smaller projects that do not require much scalability
- ▶ When you must read/write directly from the disk
- ▶ Basic development and testing

The availability of SQLite on IBM i is noteworthy because it offers a lightweight alternative to more complex database systems, providing a streamlined experience for both development and administration.

Although both SQLite and MariaDB are excellent open-source database options, they differ in architecture and feature sets. The best choice depends on the specific requirements of your project. Evaluating the strengths and limitations of each solution helps ensure the right fit for your application.

The growing availability of open-source databases, including SQLite and MariaDB, provides developers with greater flexibility. Whether selecting one of these solutions or continuing with a robust option like Db2 for IBM i, being able to choose the most appropriate tool for the job is an advantage.

9.5 Traditional programming languages

The IBM i platform has a long-standing history of reliability and continues to offer a stable foundation through traditional programming languages that are dependable in business environments.

9.5.1 RPG

RPG has long served as a cornerstone language in the IBM i ecosystem. Originally developed to generate reports, RPG evolved, especially with the introduction of ILE RPG, into a powerful tool for building complex business applications. Its efficiency in managing data-intensive tasks makes it essential for supporting core business processes.

Modernizing RPG

RPG underwent substantial changes over the years, most notably the shift from traditional fixed-form RPG to free-form RPG. Fixed-form RPG used a rigid column-based syntax that required developers to place code elements in specific columns.

- ▶ Sequence numbers in columns 1-6
- ▶ Operation codes in columns 7-11
- ▶ Statements in columns 12-80

This strict structure made the language difficult to read and maintain, especially for modern developers who are accustomed to more flexible programming languages. Despite its limitations, fixed-form RPG was highly efficient for business data processing and large-scale transaction management, which is why it remains prevalent in many IBM i systems.

The introduction of free-form RPG with RPG IV marked a significant step in modernizing the language. Free-form RPG removes the need for column-based formatting, allowing developers to write code more freely, which is similar to modern programming languages such as C, Java, or Python. Statements like **IF**, **DO**, and **FOR** are written in a more intuitive and readable format. Free-form RPG also simplifies integration with embedded SQL, enabling seamless interaction with Db2 for i databases. In addition, it supports modern programming practices, including modular design and enhanced error handling, which improve the scalability and maintainability of complex applications.

The migration to free-form RPG reflects IBM's commitment to making the IBM i platform more accessible to a new generation of developers while preserving its strengths. Free-form RPG has improved developer productivity by enabling cleaner, more readable code and enhancing the experience with modern development tools. It also supports greater integration with contemporary software practices, such as version control and IDEs, and offers a more natural approach to implementing features like data structures and error handling. As businesses modernize their systems, the adoption of free-form RPG helps ensure that IBM i remains adaptable and relevant in the evolving landscape of enterprise software development.

Another key language in the IBM i ecosystem is COBOL. Known for its clarity and effectiveness in financial and administrative functions, COBOL remains central to many systems that manage critical business data. Although often associated with older applications, COBOL's stability and reliability keep it highly relevant. The seamless operation of COBOL on IBM i underscores the platform's dedication to compatibility with earlier versions of software and the preservation of valuable business logic.

Conversion from fixed-form RPG to free-form RPG

IBM provides conversion tools that automate the migration of fixed-form RPG code to free-form RPG. These tools analyze existing code and help with the initial conversion process. However, developers must still review and test the converted code to help ensure correct function.

Converting from fixed-form RPG to free-form RPG is a valuable step toward modernizing IBM i applications. By removing column restrictions, improving code readability, and supporting contemporary development practices, free-form RPG simplifies maintenance and extension of IBM i systems. The conversion process requires understanding syntax differences and adapting the code to be more flexible, intuitive, and compatible with modern development tools. Although the process might require effort, the long-term benefits in productivity and maintainability are substantial.

9.5.2 COBOL

In addition to application-focused languages, CL plays a critical role in system administration and operational control on the IBM i OS by providing essential tools for system administrators to maintain and optimize the platform. Its value lies in its ability to streamline operations and support the efficient functioning of IBM i environments.

Together, the traditional programming languages (RPG, COBOL, and CL) shaped the IBM i platform and remain vital to many businesses. Their continued use reflects their reliability and efficiency in supporting business-critical applications. While IBM i supports modern languages, these established tools continue to provide a stable foundation for business operations.

9.6 Open-source programming on IBM i

The IBM i provides a wealth of open-source programming languages that can complement and enhance traditional programming languages.

Figure 9-39 shows the timeline for when open-source programming languages and frameworks were first added to IBM i.

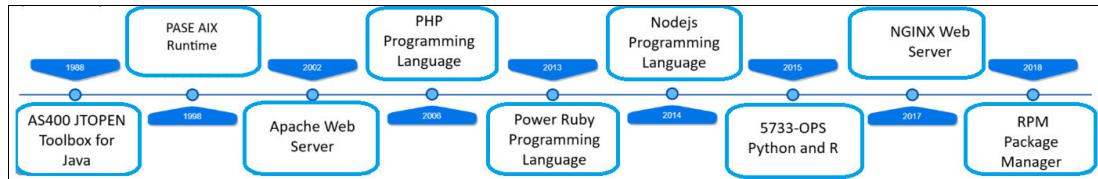


Figure 9-39 Open-source timeline on IBM i

9.6.1 Available languages

The following open-source languages are available on the IBM i platform.

Node.js

Node.js is an open-source, cross-platform JavaScript runtime environment that runs the JavaScript 8 engine outside the browser. Node.js was introduced to IBM i in 2014 as part of the 5733-OPS licensed program and later migrated to the RPM-based installation process.

Python

Python is a high-level, interpreted programming language that is known for its readability and simplicity. It is a widely used language in modern software development. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming, making it a flexible choice for a wide range of applications.

PHP

PHP is an open-source scripting language for web development. It is a foundational technology of the web. Embedded within HTML, PHP is suited for server-side development, enabling the creation of dynamic web pages and applications. It is known for its versatility and integration with various databases, including MySQL, PostgreSQL, and SQLite.

Here is a timeline of PHP on IBM i:

2002	Unsupported versions of PHP on IBM i.
2006	Zend release a version of their Zend Server product that was installed as a licensed program. Apart from the basic license, it is a chargeable feature.
2018	IBM and Zend released community PHP, which is an unsupported open-source offering that is installed through the RPM package manager.

Ruby

Ruby is a dynamic, open-source programming language that is best known for its web application framework, Ruby on Rails, which contributed to the language's popularity. Ruby has been available on IBM i since 2013 and was originally distributed through the 5773-OPS licensed program.

R is a software environment for statistical computing and graphics. It is suitable for performing data analysis to generate business insights. R has been available on IBM i since 2015, also through the 5773-OPS licensed program. However, it is not widely used on IBM i and receives limited support from IBM.

Java

Java is a high-level, object-oriented programming language that is developed by Sun Microsystems (now part of Oracle Corporation). It is known for its “write once, run anywhere” (WORA) capability. Java has been available on IBM i since the introduction of the AS/400 in 1988, initially through the JTOpen Toolbox for Java, which was distributed as a traditional licensed program.

9.6.2 Open-source programming support

IBM supports the following programs and products:

- ▶ 5733-SC1 OpenSSL and OpenSSH
- ▶ 5770-DG1 Apache HTTP Server and XML service
- ▶ 5770-JV1 Java

There is no longer any support for the original method of supplying open-source from the 5733-OPS licensed program,

IBM partially supports these integration packages:

- ▶ iToolkit
- ▶ idb-connector
- ▶ idb-pconnector
- ▶ loopback-connector-IBM i
- ▶ ODBC

These Python packages are also supported:

- ▶ toolkit
- ▶ ibm_db

For more information, see [Open Source Support for IBM i](#).

9.7 Integrating traditional and open-source languages

This section explores how to integrate traditional programming languages on the IBM i platform with open-source languages. Combining these approaches enables organizations to use the strengths of both environments.

Although IBM introduced several open-source languages, their adoption often depends on integrating them effectively with traditional IBM i languages. This integration enables enterprises to maximize the benefits of modern technologies while maintaining the stability of their existing systems.

In today's rapidly evolving technology landscape, combining traditional and open-source languages on IBM i is essential. This approach enables businesses to preserve past investments while embracing the flexibility and capabilities of open-source solutions. For organizations seeking enhanced functions, adaptability, and modernization, this integration is a critical strategy.

Traditional languages connecting to open-source languages

Here are some examples of traditional languages that use open-source extensions:

- ▶ Using CL to do the following functions:
 - Call a Node.js script.
 - Call a PHP script.
 - Call a Python script.
- ▶ Using free-form RPG to do the following functions:
 - Call a Node.js script.
 - Call a PHP script.
 - Call a Python script.

Open-source languages connecting to traditional languages

Here are some examples of writing new, open-source routines that connect to traditional languages:

- ▶ Using Node.js to do the following functions:
 - Call service programs.
 - Run IBM i commands.
- ▶ Using PHP to do the following functions:
 - Call service programs.
 - Run IBM i commands.
- ▶ Using Python to do the following functions:
 - Call CL programs.
 - Run IBM i commands.

The ability to combine new technologies with existing application code enables organizations to extend and modernize systems. By integrating newer programming languages, businesses can revitalize their applications while preserving core functions. This approach also enables teams to use the skills of developers who might not be familiar with traditional IBM i languages, increasing flexibility and fostering innovation in application development.

Important considerations

All examples in this section use environment variables to store sensitive information, helping ensure that security-related values are not exposed. This approach supports practical demonstrations while maintaining best practices in data protection.

The section focuses exclusively on the latest version of free-form RPG IV, which is the preferred version for modernization tasks and projects.

Language versions and terminology

All code snippets in this section were verified with the languages that are shown in Table 9-4. This section refers to each language with the short name that is shown in the table.

Table 9-4 Programming languages covered

Name	Short name	Version
CL	CL	7.5
Free-form RPG IV	RPG	7.5
Node.js	Node	22.13.1-1
PHP84	PHP	8.4.4
Python3	Python	3.9.21-1

9.7.1 Traditional languages to open-source languages

This section provides connectivity from traditional languages to open-source routines.

Command Language or Control Language

This section describes using IBM i CL to call open-source routines.

CL to Node.js

The first example in the CL section demonstrates a simple CL program that runs a Node.js script. It uses the **QSH** (Qshell) command to start a shell session. **QSH** is an IBM i command that is built in to the OS.

Example 9-14 shows the CL script.

Example 9-14 CL script to call the Node.js hello world script

```
PGM

DCL      VAR(&CMD) TYPE(*CHAR) LEN(200)

/* Define the command to run the Node.js script */
CHGVAR   VAR(&CMD) VALUE('/QOpenSys/pkg/bin/node +
                           /home/user/node1.js')

/* Call the Node.js script */
QSH      CMD(&CMD)

ENDPGM
```

The Node.js script is stored in the IBM i IFS (the file name is included in the variable <VALUE> in the CL script).

Example 9-15 on page 329 shows the Node.js script contents.

Example 9-15 The Node.js script that is called by CL

```
console.log("Hello from Node.js!");
```

The script outputs “Hello from Node.js!” to the job log, which proves that you can run scripts within your IFS.

CL to PHP

This example builds on the concept in “CL to Node.js” by using parameters that are passed from the CL program to the PHP script. Parameters are often called arguments or options in open source.

Example 9-16 shows the CL program.

Example 9-16 CL to call the PHP routine

```
PGM      PARM(&ARG1 &ARG2)
DCL      VAR(&ARG1) TYPE(*CHAR) LEN(50)
DCL      VAR(&ARG2) TYPE(*CHAR) LEN(50)
DCL      VAR(&CMD) TYPE(*CHAR) LEN(300)

/* Construct the command to run the PHP script with arguments */
CHGVAR   VAR(&CMD) VALUE('/QOpenSys/pkgs/bin/php +
/home/user/php1.php ' *CAT &ARG1 *CAT ' ' +
*CAT &ARG2)

/* Call the PHP script */
QSH      CMD(&CMD)

ENDPGM
```

Example 9-17 shows the PHP routine.

Example 9-17 PHP routine

```
<?php

// Retrieve the arguments from the command line
$arg1 = $argv[1];
$arg2 = $argv[2];
// Display the arguments (or perform operations with them)
echo "Argument 1: $arg1\n";
echo "Argument 2: $arg2\n";
// Perform operations with the arguments
$result = $arg1 . ' ' . $arg2;
echo "Result: $result\n";
```

Example 9-18 shows how to call the CL program.

Example 9-18 Calling the CL program

```
CALL CALLPHP PARM('Hello' 'World')
Argument 1: hello
Argument 2: world
Result: hello world
```

CL to Python

This Python language example follows on from the ones in “CL to Node.js” on page 328 and “CL to PHP” on page 329.

Example 9-19 show the CL script.

Example 9-19 CL script to call the Python routine

```
PGM      PARM(&ARG1 &ARG2)
DCL      VAR(&ARG1) TYPE(*CHAR) LEN(50)
DCL      VAR(&ARG2) TYPE(*CHAR) LEN(50)
DCL      VAR(&CMD) TYPE(*CHAR) LEN(300)

/* Construct the command to run the Python script with arguments */
CHGVAR   VAR(&CMD) VALUE('/QOpenSys/pkgs/bin/python +
/home/user/python.py ' *CAT &ARG1 *CAT ' ' +
*CAT &ARG2)

/* Call the PHP script */
QSH      CMD(&CMD)

ENDPGM
```

Example 9-20 shows the Python routine.

Example 9-20 Python routine that is called from CL

```
# This script is used to demonstrate how to pass arguments to a Python script
# from the command line.
import sys
# Retrieve arguments from the command line
arg1 = sys.argv[1]
arg2 = sys.argv[2]
# Display the arguments (or perform operations with them)
print(f"Argument 1: {arg1}")
print(f"Argument 2: {arg2}")
```

Example 9-21 the output of the Python routine.

Example 9-21 Output from the Python routine

```
Argument 1: hello
Argument 2: world
```

RPG

The examples in this section were created by using RPG to call different open-source routines.

RPG calling a node script

Example 9-22 shows how a simple RPG program can call a Node.js script.

Example 9-22 RPG code to call a Node.js script

```
**free
// Prototype for QCMDexc API
dcl-pr QCMDexc extpgm('QCMDexc');
CmdString    char(3000) const; // Command string to run
```

```

    CmdLength  packed(15:5) const; // Length of the command string
end-pr;
// Main program
dcl-s Cmd char(3000); // Variable to hold the command string
// Build the QSH command to call the Node.js script
Cmd = 'QSH CMD('''/QOpenSys/pkg/bin/node /home/user/node1.js''')';
// Run the command by using QCMDexc
QCMDexc(Cmd: %len(%trim(Cmd)));
*inlr = *on; // End program

```

Example 9-23 shows the node script that is called.

Example 9-23 The node.js script that is called

```

// node1.js
console.log("Hello from Node.js!");

```

RPG calling a PHP script

Example 9-24 shows RPG calling a PHP script.

Example 9-24 RPG calling PHP script

```

**free
// Input Parameters
dcl-pi *n;
    Param1 char(1) const;
    Param2 char(1) const;
end-pi;
dcl-pr QCMDexc extpgm('QCMDexc');
    CmdString char(32767) const options(*varsize);
    CmdLength packed(15:5) const;
end-pr;
dcl-s CmdString1 char(1000);
dcl-s CmdString2 char(1000);
dcl-s CmdLength packed(15:5);
dcl-s quote varchar(10) inz(''''');
// Mess about building the QSH command
CmdString1 =
    '/QOpenSys/pkg/bin/php /home/user/redbook/php1.php ' +
    %trim(Param1) + ' ' + %trim(Param2);
cmdstring2 = 'QSH cmd(' + quote + %trim(cmdstring1) + quote + ')';
CmdLength = %len(%trimr(CmdString2));
// Running the command
QCMDexc(CmdString2: CmdLength);
*inlr = *on; // End program 'omers

```

Example 9-25 shows the PHP script that is called by the RPG routine.

Example 9-25 PHP script

```

<?php

// Retrieve the arguments from the command line
$arg1 = $argv[1];
$arg2 = $argv[2];
// Display the arguments (or perform operations with them)

```

```
echo "Argument 1: $arg1\n";
echo "Argument 2: $arg2\n";
// Perform operations with the arguments
$result = $arg1 . ' ' . $arg2;
echo "Result: $result\n";
```

RPG calling a Python script

In this example, the RPG program constructs a command string to call a Python script by using the **QSH** command. The command passes the parameters **param1** and **param2** to the script. The Python script retrieves these parameters from the CLI arguments and processes them.

Example 9-26 shows the RPG code.

Example 9-26 RPG code to call the Python script

```
**free
// Input Parameters
dcl-pi *n;
    Param1 char(1) const;
    Param2 char(1) const;
end-pi;
dcl-pr QCMDexc extpgm('QCMDexc');
    CmdString char(32767) const options(*varsize);
    CmdLength packed(15:5) const;
end-pr;
dcl-s CmdString1 char(1000);
dcl-s CmdString2 char(1000);
dcl-s CmdLength packed(15:5);
dcl-s quote varchar(10) inz('\'');
// Mess about building the QSH command
CmdString1 =
    '/QOpenSys/pkgs/bin/python /home/user/redbook/python1.py ' +
    %trim(Param1) + ' ' + %trim(Param2);
cmdstring2 = 'QSH cmd(' + quote + %trim(cmdstring1) + quote + ')';
CmdLength = %len(%trimr(CmdString2));
// Running the command
QCMDexc(CmdString2: CmdLength);
*inlr = *on; // End program 'omers
```

Example 9-27 shows the Python script.

Example 9-27 Python script that is called by CL

```
# This script is used to demonstrate how to pass arguments to a Python script
# from the command line.
import sys
# Retrieve arguments from the command line
arg1 = sys.argv[1]
arg2 = sys.argv[2]
# Display the arguments (or perform some operations)
print(f"Argument 1: {arg1}")
print(f"Argument 2: {arg2}")
```

9.7.2 Open-source languages to traditional languages

This section provides examples of calling traditional IBM i routines from new, open-source language routines.

IBM i Node.js

The IBM i Node.js iToolkit is an interface that simplifies the connection between modern web applications and the IBM i platform. It enables traditional systems to create web services and applications that interact seamlessly with IBM i data and processes. With iToolkit, developers can streamline operations and explore new opportunities for innovation on IBM i.

To install this toolkit, run the following Node Package Manager (NPM) command:

```
npm install itoolkit
```

After you install the toolkit, create a connection to your IBM i. Here are the supported transport methods for creating a connection:

ODBC	Use the ODBC transport to call Db2 for i stored procedures. This service is a generic connection and can also be used to other (third-party) servers.
SSH	Use the SSH transport to make an SSH connection.
idb	Use this service as an alternative to the ODBC system. You can use it only for connections to an IBM i server.
REST	This connection makes an HTTP request to a REST API endpoint.

All examples in the remainder of this section use the SSH transport method. If you prefer or need to connect by using a different transport method, see the IBM [open-source documentation](#).

Node calling a service program

This example demonstrates how Node.js can be integrated with traditional IBM i service programs, including ones that are written in RPG, COBOL, or CL. In this example, a Node.js program calls an IBM service program that is named **QC2UTIL2**. This service program is part of the IBM i OS and is available across all supported software versions.

As shown in Figure 9-40, **QC2UTIL2** includes multiple procedures that are exported and accessible to other programs. This example uses the **acos** procedure to illustrate how parameters can be passed to the service program and how a return value can be received.

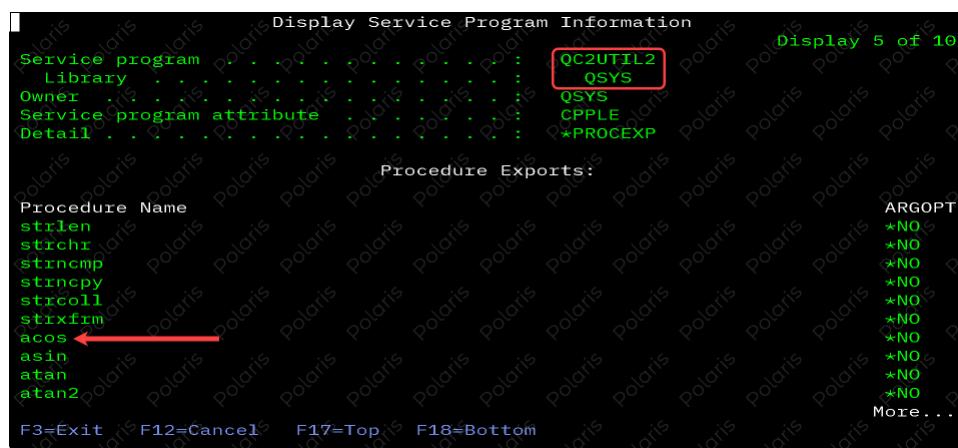


Figure 9-40 Capabilities of QC2UTIL2

This example uses the fast-xml-parser module to convert the raw XML output from iToolkit into JSON, which can be read natively by the Node.js script. This module is automatically installed as a dependency of the latest iToolkit package, and NPM handles the installation process.

Example 9-28 shows the Node.js script.

Example 9-28 The Node.js program

```
/*
 * This script demonstrates how to use the itoolkit library to call an IBM i program.
 * It establishes an SSH connection to an IBM i system by using credentials from environment
variables,
 * and then calls the 'cos' function in the QC2UTIL2 program in the QSYS library.
 * The script adds a parameter and a return value to the program call, runs the connection,
 * and parses the XML output to log the return data.
 */
const { Connection, ProgramCall } = require('itoolkit'); // Import Connection and ProgramCall
from itoolkit
const { XMLParser } = require('fast-xml-parser'); // Import XMLParser from fast-xml-parser
require('dotenv').config(); // Load environment variables from .env file
// Create a new connection by using SSH transport with credentials from environment variables
const conn = new Connection({
  transport: 'ssh',
  transportOptions: { host: process.env.IBMI_HOST, username: process.env.IBMI_USER, password:
process.env.IBMI_PASSWORD },
});
// Create a new program call to the QC2UTIL2 program in the QSYS library, calling the 'cos'
function
const program = new ProgramCall('QC2UTIL2', { lib: 'QSYS', func: 'cos' });
program.addParam({ type: '8f', value: '0', by: 'val' }); // Add a parameter of type '8f' with
value '0'
program.addReturn({ type: '8f', value: '' }); // Add a return value of type '8f'
// Add the program call to the connection
conn.add(program);
conn.debug(true); // Enable debugging for the connection
// Run the connection and handle the response
conn.run((error, xmlOutput) => {
  if (error) {
    throw error; // Throw an error if the connection fails
  }
  const Parser = new XMLParser(); // Create a new XML parser
  const result = Parser.parse(xmlOutput); // Parse the XML output
  console.log(result.myscript.pgm.return.data); // Log the return data from the parsed XML
});
```

Example 9-29 shows the results from running this example program.

Example 9-29 Program output

```
=====
INPUT XML
=====
<?xml version='1.0'?><myscript><pgm name='QC2UTIL2' lib='QSYS' func='cos' error='fast'><parm
by='val'><data type='8f'>0</data></parm><return><data
type='8f'></data></return></pgm></myscript>
```

```

SSH Client is ready
STDOUT:
<?xml version='1.0'?><myscript><pgm name='QC2UTIL2' lib='QSYS' func='cos' error='fast'>
<parm by='val'>
<data type='8f'>0</data>
</parm>
<return>
<data type='8f'>1</data>
</return>
<success><![CDATA[+++ success QSYS QC2UTIL2 cos ]]></success>
</pgm>
</myscript>
stdin has ended
stdout has ended
Stream exit code: 0
=====
OUTPUT XML
=====
<?xml version='1.0'?><myscript><pgm name='QC2UTIL2' lib='QSYS' func='cos' error='fast'>
<parm by='val'>
<data type='8f'>0</data>
</parm>
<return>
<data type='8f'>1</data>
</return>
<success><![CDATA[+++ success QSYS QC2UTIL2 cos ]]></success>
</pgm>
</myscript>
1
SSH Client has ended
SSH Client has closed

```

Note: If **debug** is disabled, you see only the return value of 1.

Node.js running IBM i commands

This example demonstrates how to run an IBM i command by using the Node.js iToolkit. The command retrieves the current job's library list, including both user and system portions. The command that is used is **RTVJOBA**.

Example 9-30 shows the Node.js program.

Example 9-30 Node.js program to run RTVJOBA

```

/**
 * This script demonstrates how to use the itoolkit library to run a CL command on an IBM i
 * system.
 * It establishes an SSH connection to an IBM i system by using credentials from environment
 * variables,
 * and then runs the 'RTVJOBA' CL command to retrieve the user and system library lists.
 * The script parses the XML output and logs the result as a JSON string.
 */
const { Connection, CommandCall } = require('itoolkit'); // Import Connection and CommandCall
from itoolkit
const { XMLParser } = require('fast-xml-parser'); // Import XMLParser from fast-xml-parser
require('dotenv').config(); // Load environment variables from .env file

```

```

// Create a new connection by using SSH transport with credentials from environment variables
const conn = new Connection({
  transport: 'ssh',
  transportOptions: { host: process.env.IBMI_HOST, username: process.env.IBMI_USER, password: process.env.IBMI_PASSWORD },
});
// Create a new command call to run the 'RTVJOBA' CL command
const command = new CommandCall({ type: 'cl', command: 'RTVJOBA USRLIBL(?) SYSLIBL(?)' });
// Add the command call to the connection
conn.add(command);
conn.debug(true); // Enable debugging for the connection
// Run the connection and handle the response
conn.run((error, xmlOutput) => {
  if (error) {
    throw error; // Throw an error if the connection fails
  }
  const Parser = new XMLParser(); // Create a new XML parser
  const result = Parser.parse(xmlOutput); // Parse the XML output
  console.log(JSON.stringify(result)); // Log the result as a JSON string
});

```

Example 9-31 shows the results.

Example 9-31 Results of running the program

```

{
  "?xml": "",
  "myscript": {
    "cmd": {
      "success": "+++ success RTVJOBA USRLIBL(?) SYSLIBL(?)",
      "row": [
        {
          "data": "QTEMP      QGPL      POWERWIRE  F_MTD_SRC  TOOLS      FORMASERVE  UNIXCMD
YAJL   F_MTD_DEV  QSHONI"
        },
        {
          "data": "QSYS      QSYS2      QHLPSYS    QUSRSYS"
        }
      ]
    }
  }
}

```

Python

In this section, you use Python to connect to traditional IBM i functions.

Calling programs

This example demonstrates how to use Python to call an RPG program with two parameters: one for input and one for output. The output is returned to the Python script for further processing. The Python code, which is shown in Example 9-32 on page 337, includes basic error handling to help ensure reliability.

Example 9-32 Example Python code to run an RPG program

```
from itoolkit import iToolKit, iCmd, iPgm, iData # Importing necessary modules from the
itoolkit package
from itoolkit.transport import DatabaseTransport # For database communication
import ibm_db_dbi # For working with the IBM Db2 database
try:
    # Establishing a connection to the IBM Db2 database
    # Replace with appropriate parameters if required (for example, host, username, password)
    conn = ibm_db_dbi.connect()
    # Initializing the iToolKit and setting up the database transport
    itransport = DatabaseTransport(conn)
    itool = iToolKit()
    # Adding a library to the library list
    # This ensures that the program can access required library resources
    itool.add(iCmd('addlib', 'addlib redbook'))
    # Adding the RPG program to be called and configuring its parameters
    itool.add(
        iPgm('rpgResults', 'RPG7') # Specifying the RPG program name
        .addParm(iData('parm1_input', '10a', 'James')) # Input parameter: 10-character string
        with value 'James'
        .addParm(iData('parm2_output', '10a', ' ')) # Output parameter: 10-character string,
        initialized as empty
    )
    # Running the program call through the toolkit
    itool.call(itransport)
    # Retrieving the program's output as a dictionary
    rpgResults = itool.dict_out('rpgResults')
    # Displaying the full results of the RPG program call
    print("RPG Results:", rpgResults)
    # Checking if the RPG program ran successfully
    if rpgResults.get('success'):
        print("Success!") # Indicating success
    else:
        print("Errors occurred.") # Indicating an error in the program execution
    # Displaying the input and output parameters along with the program's execution status
    print("\nRPG input parameter:", rpgResults.get('parm1_input', 'N/A')) # Input parameter
    print("RPG output:", rpgResults.get('parm2_output', 'N/A')) # Output parameter
    print("Call status:", rpgResults.get('success', 'N/A')) # Execution status
except Exception as e:
    # Handling any exceptions that occur during script execution
    print("An error occurred:", str(e))
```

Example 9-33 shows the RPG source.

Example 9-33 RPG source

```
**free
// Take one parameter from a Python program and return another
// Nothing fancy, but it works!
// Parameters
dcl-pi *n;
    Arg1 char(10) const;
    Result char(10);      // Output parameter
end-pi;
If %trim(Arg1) = 'James';
```

```
    Result = 'Riddle';
Else;
    Result = 'Unknown' ;
Endif;
*inlr = *on; // End program 'omers
```

Example 9-34 shows the output from running the Python program.

Example 9-34 Output from the RPG program

```
RPG Results: {'parm1_input': 'James', 'parm2_output': 'Riddle', 'success': '+++'
success  RPG7'}
Success!
RPG input parameter: James
RPG output: Riddle
Call status: +++ success  RPG7
```

Using Python to run IBM i commands

This example shows a typical Python script that runs an IBM i command. The script establishes a database connection and then calls the **QCMDDEXC** stored procedure to run the command on the IBM i system.

Example 9-35 shows the Python code.

Example 9-35 The Python code to run an IBM i command

```
# This is a simple example of how to call a stored procedure in Db2 for i by using
# the ibm_db_dbi module.
from ibm_db_dbi import connect
try:
    # Connect to the database
    conn = connect()
    # Create a cursor
    cur = conn.cursor()

    # Call the stored procedure
    cur.callproc('qcmdexc', ('SNDMSG HELLO ANDY',))
    # Close the cursor
    cur.close()
except Exception as e:
    # Handle exceptions
    print(f"Error: {e}")
```

This example demonstrates an alternative method for running an IBM i command. It uses iToolkit to establish a database connection, which is then used to run commands on the IBM i system.

Example 9-36 shows the code.

Example 9-36 Example Python code to run the WRKACTJOB command

```
from itoolkit import *
from itoolkit.transport import DatabaseTransport
import ibm_db_dbi
from itoolkit import *
from itoolkit.transport import DatabaseTransport
```

```

import ibm_db_dbi
try:
    # Establish a connection to the IBM i system
    conn = ibm_db_dbi.connect()
    itransport = DatabaseTransport(conn)
    itool = iToolKit()
    # Add and run the WRKACTJOB command
    itool.add(iCmd5250('wrkactjob', 'WRKACTJOB'))
    itool.call(itransport)
    # Retrieve and print the output of the WRKACTJOB command
    wrkactjob = itool.dict_out('wrkactjob')
    print(wrkactjob)
except Exception as e:
    # Handle exceptions
    print(f"Error: {e}")

```

PHP

In this section, you use PHP to connect to an IBM i system and run commands and programs.

PHP calling an RPG program

This section provides an example of how a PHP script can call an IBM i RPG program. Communication is handled by using the Toolkit for IBM i, which is a PHP library that offers a straightforward interface for calling programs and accessing IBM i resources. The toolkit must be installed on the IBM i system before using any of the examples in this section.

The Toolkit is installed through the Open Source Package Management utility, which is included as part of the IBM ACS package.

Figure 9-41 shows the Toolkit.

php-openssl	7.3.13-0	@r
php-pdo	7.3.13-0	@r
php-phar	7.3.13-0	@r
php-process	7.3.13-0	@r
php-soap	7.3.13-0	@r
php-sockets	7.3.13-0	@r
php-sodium	7.3.13-0	@r
php-sqlite3	7.3.13-0	@r
php-tokenizer	7.3.13-0	@r
php-xml	7.3.13-0	@r
php83zend	8.3-2	@r
php83zend-php-cli	8.3.4-0	@r
php83zend-php-common	8.3.4-0	@r
php83zend-php-opcache	8.3.4-0	@r
php83zend-php-pecl-ibm-db2	2.2.0-0	@r
php83zend-php-xml	8.3.4-0	@r
phpzend-apache-conf	1.1-1	@r
phpzend-xmikit	1.8.5-1	@r
pigz	2.4-1	@r
postgresql12	12.2-2	@r
postgresql12-contrib	12.2-2	@r
postgresql12-devel	12.2-2	@r
postgresql12-docs	12.2-2	@r
postgresql12-libpq	12.2-2	@r

Figure 9-41 List of programs in IBM i Open Source Package Management

Example 9-37 shows the PHP source.

Example 9-37 The PHP source for running RPG

```
<?php
require_once 'ToolkitService.php'; // Include the IBM i Toolkit library
try {
    // Establish a connection to the IBM i system by using environment variables
    $options = array(
        'database' => '*LOCAL', // Database name (usually default is '*LOCAL')
        'user' => '', // IBM i user profile
        'password' => '' // IBM i password
    );
    // Create a toolkit service object
    $conn = ToolkitService::getInstance('*LOCAL', '', '');
    // specify stateless mode (simple: no internal key needed)
    $conn->setOptions(array('stateless' => true));
    // Define parameters for the RPG program
    $Params = [
        ['name' => 'ARG1', 'io' => 'in', 'type' => '10A', 'value' => 'James']
    ];
    // Call the RPG program
    $outputParams = $conn->PgmCall('RPG7', 'Redbooks', $Params);
    // Display the result
    echo 'Result: ' . $outputParams['RESULT'] . "\n";
} catch (Exception $e) {
    // Handle exceptions
    echo 'Error: ' . $e->getMessage() . "\n";
}
```

The RPG program that is called by the PHP script in this example is the same program that is used in the Python integration example, as shown in Example 9-33 on page 337.

PHP running a IBM i command

This example demonstrates how a PHP script can run an IBM i command. It uses the database transport mechanism that is provided by the PHP iToolkit to run the **WRKACTJOB** command.

Example 9-38 shows the PHP source.

Example 9-38 The PHP source for calling the IBM i command

```
<?php
// This script connects to an IBM i system by using the iToolkit and retrieves the
// output of the WRKACTJOB command.
// Enable error reporting for development purposes.
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);
// Include the iToolkit library.
from itoolkit import *
from itoolkit.transport import DatabaseTransport
import ibm_db_dbi
// Connect to the IBM i system by using the ibm_db_dbi library.
conn = ibm_db_dbi.connect()
itransport = DatabaseTransport(conn)
```

```

// Create an iToolKit instance and add the WRKACTJOB command.
itool = iToolkit()
itool.add(iCmd5250('wrkactjob', 'WRKACTJOB'))
// Call the command and retrieve the output.
// Note: The command runs in a 5250 session, so the output will be in a format
// suitable for display on a 5250 terminal.
// The output is not directly usable in a web application, but you can parse it as
// needed.
itool.call(itransport)
wrkactjob = itool.dict_out('wrkactjob')
print(wrkactjob)

```

Figure 9-42 shows the output of the command.

Work with Active Jobs													Page 1	
													POLARIS 18/03/25 11:29:02 GMT	
Subsystem/Job	User	Number	User	Type	Pool	Pty	CPU	Int	Rsp	AuxIO	CPU%	Function	Status	Threads
QBATCH	QSYS	096366	QSYS	SBS	2	0	.0	0	0	0	0	DEQW	2	
QCMN	QSYS	096367	QSYS	SBS	2	0	.1	0	0	0	0	DEQW	2	
QACSTOP	QUSER	096394	QUSER	PJ	2	20	.0	0	0	0	0	PSRW	1	
QLZPSERV	QUSER	096416	QUSER	PJ	2	20	.0	0	0	0	0	PSRW	1	
QNMAPINGD	QUSER	096388	QUSER	PJ	2	25	.0	0	0	0	0	PSRW	1	
QNMAPEXCD	QUSER	096391	QUSER	PJ	2	25	.0	0	0	0	0	PSRW	1	
QNPSERVER	QUSER	096412	QUSER	PJ	2	20	.0	0	0	0	0	PSRW	1	
QZCSRVR	QUSER	096401	QUSER	PJ	2	20	.0	0	0	0	0	PSRW	1	
QZCCSRVR	QUSER	096408	QUSER	PJ	2	20	.0	0	0	0	0	PSRW	1	
QCTL	QSYS	096334	QSYS	SBS	2	0	.1	0	0	0	0	DEQW	2	
QSYSSGD	QPGRMR	096357	QPGRMR	BCH	2	10	.1	0	0	0	0	EVTW	1	
QHTTPPSVR	QSYS	096459	QSYS	SBS	2	0	.0	0	0	0	0	DEQW	2	
ADMIN	QTMHHTTP	096460	QTMHHTTP	BCH	2	25	2.9	0	0	0	0	PGM-QZHBMAIN	SIGW	
ADMIN	QTMHHTTP	096470	QTMHHTTP	BCI	2	25	104.1	0	0	0	0	PGM-QZSRLOG	SIGW	
ADMIN	QTMHHTTP	096480	QTMHHTTP	BCI	2	25	101.4	0	0	0	0	PGM-QZSRHTTP	SIGW	
ADMIN	QTMHHTTP	506700	QSECOFR	BCI	2	25	.0	0	0	0	0	PGM-QZSRCGI	TIMW	
ADMIN1	QWEBADMIN	096481	QWEBADMIN	BCI	2	25	1076.0	0	0	0	0	JVM-/QIBM/Prod	THDW	
ADMIN2	QLWISVR	096483	QLWISVR	BCI	2	25	1308.0	0	0	0	0	JVM-/QIBM/Prod	THDW	
ADMIN3	QWEBADMIN	096482	QWEBADMIN	BCI	2	25	50.9	0	0	0	0	JVM-/QIBM/Prod	THDW	
ADMIN4	QWEBADMIN	096484	QWEBADMIN	BCI	2	25	122.1	0	0	0	0	JVM-/QIBM/Prod	THDW	
ADMIN5	QLWISVR	096485	QLWISVR	BCI	2	25	2191.8	0	0	0	0	JVM-/QIBM/Prod	THDW	
COMPANY	ANDY	506787	ANDY	BCI	2	25	589.3	0	0	0	0	JVM-/QIBM/Prod	THDW	
COMPANY	QTMHHTTP	506782	QTMHHTTP	BCH	2	25	1.6	0	0	0	0	PGM-QZHBMAIN	SIGW	
COMPANY	QTMHHTTP	506785	QTMHHTTP	BCI	2	25	58.5	0	0	0	0	PGM-QZSRLOG	SIGW	
COMPANY	QTMHHTTP	506794	QTMHHTTP	BCI	2	25	56.9	0	0	0	0	PGM-QZSRHTTP	SIGW	
INTRANET	QTMHHTTP	096743	QTMHHTTP	BCH	2	25	2.8	0	0	0	0	PGM-QZHBMAIN	SIGW	

Figure 9-42 Output from WRKACTJOB



Linux and Red Hat OpenShift

Linux on IBM Power combines the open-source flexibility of Linux with the performance, reliability, and scalability of the IBM Power Architecture. This combination delivers substantial benefits for various workloads.

IBM began supporting Linux in the late 1990s as the operating system (OS) gained influence in the technology industry. In 2000, IBM publicly committed to Linux, and in 2001, it invested \$1 billion to advance Linux on IBM Power. This initiative included optimizing the Linux kernel and software to run efficiently on IBM hardware. Today, various Linux distributions, such as Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server, and Ubuntu, are supported on IBM Power servers, with custom enhancements that are tailored for the IBM Power Architecture. IBM also collaborates closely with these vendors to help ensure consistent and well-supported OS environments.

For organizations that require high-performance, reliable, scalable, and secure platforms for critical workloads, Linux on IBM Power provides a compelling solution. Its long history of innovation and open collaboration continues to make it a strong choice in the evolving computing landscape.

Linux provides a strong foundation for modern workloads and powers Red Hat OpenShift, which is a Kubernetes-based container platform for deploying hybrid cloud applications on IBM Power. By using both Linux and Red Hat OpenShift, organizations can run hybrid cloud applications on the hardware platform that best meets their specific requirements.

The following topics are described in this chapter:

- ▶ 10.1, “Linux” on page 344
- ▶ 10.2, “Containerization solutions on IBM Power” on page 351

10.1 Linux

Linux is an open-source, cross-platform OS that runs on many platforms, from embedded systems to mainframe computers. It provides a UNIX like implementation across multiple computer architectures.

IBM Power servers offer unique capabilities for running enterprise Linux distributions with a fully open stack that benefits from the OpenPower ecosystem and delivers efficient cloud-native performance through IBM PowerVM virtualization technology. These servers enhance the reliability, security, and scalability of open-source technology with industry-leading, cloud-native deployment options. Enterprise Linux on IBM Power provides a solid foundation for open-source, hybrid cloud infrastructure, enabling organizations to modernize applications more efficiently.

Linux on IBM Power provides you with the blend of agility, security, and resiliency that you need to stay ahead.

Figure 10-1 shows some of the benefits of running Linux on IBM Power.

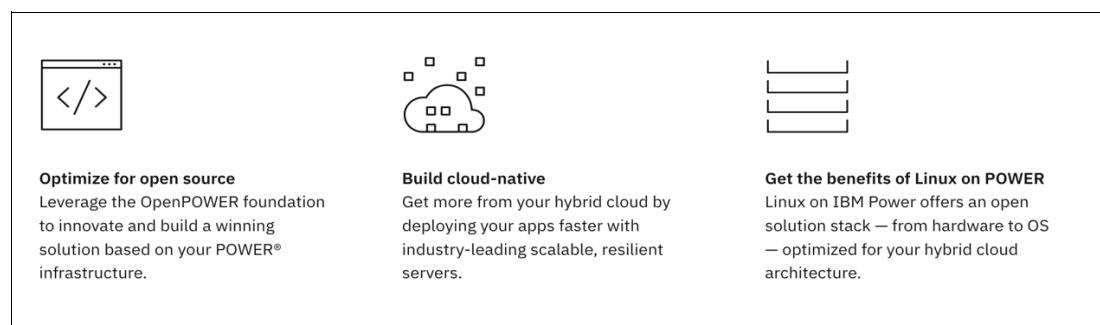


Figure 10-1 Benefits of Linux on Power

Application modernization is the process of updating an application so that it can be maintained, extended, deployed, and managed in a way that meets current and future requirements. On IBM Power, organizations can continue running existing applications while integrating new cloud-native applications alongside them.

Application modernization opens the door to several business and technical benefits for your organization. Look closer at some of them:

- ▶ Accelerate digital transformation.

Organizations increasingly seek new ways to deliver innovative and engaging experiences that retain existing customers, attract new ones, and gain a competitive edge. A Forrester Consulting study that was commissioned by IBM found that modernizing applications with IBM and Red Hat solutions can accelerate release frequency by up to ten times¹, improving customer engagement, time to market, and operational efficiency.

- ▶ Gain a superior developer experience.

An organization's most valuable assets are its people. To uncover hidden competitive advantages through IT, help ensure that application developers have access to the right technologies and the most up-to-date applications. This access enables them to innovate and deliver exceptional customer experiences.

¹ Source:

https://www.redhat.com/tracks/_pfcdn/assets/10330/contents/211817/273b8203-4ba5-499f-a858-f276b058da df.pdf

- ▶ Deploy enterprise applications across your hybrid multicloud.

As enterprises continue to adopt hybrid cloud strategies, applications must have the flexibility to be deployed across diverse environments to realize their full benefits. This flexibility enables organizations to use ongoing innovation from public cloud providers while maintaining the security, data privacy, and reliability of their own data centers. This level of choice and flexibility is critical for achieving competitive differentiation in today's market.

10.1.1 Linux distributions that are supported on IBM Power

IBM Power supports a range of Linux distributions, each offering distinct features and capabilities. This section provides an overview of the supported distributions.

Red Hat-based distributions

RHEL and its derivatives (CentOS, Fedora, AlmaLinux, and Rocky Linux) are widely used distributions that are known for their stability, security, and enterprise-grade support. These distributions support a broad range of use cases, from traditional workloads such as application servers and large databases to modern environments including containers and machine learning (ML).

Running RHEL on IBM Power servers enables users to use a robust enterprise Linux distribution on high-performance IBM Power hardware. This combination is ideal for demanding workloads such as big data processing and large-scale virtualization, offering the flexibility of open-source Linux alongside the reliability and performance of IBM Power architecture.

RHEL CoreOS, which is based on RHEL, represents the next generation of single-purpose container OS technology. It maintains the same quality standards as RHEL and includes automated, remote upgrade capabilities. RHEL CoreOS is supported exclusively as a component of the Red Hat OpenShift Container Platform and is used for all machines within that platform.

Fedora CoreOS is a minimal, monolithic, and container-focused OS that updates automatically. Although designed for cluster environments, it can also operate as a stand-alone OS. Fedora CoreOS is optimized for Kubernetes and serves as a secure, scalable container host for running containerized workloads.

For more information about RHEL, see this [Red Hat website](#).

SUSE-based distributions

SUSE Linux Enterprise Server, traditionally used for SAP HANA on IBM Power environments, also supports classic workloads. In addition, openSUSE Leap is a community-driven, open-source Linux distribution that is developed by the openSUSE Project. It shares its core with SUSE Linux Enterprise, providing a highly stable and well-tested foundation. openSUSE Leap receives the same security fixes as SUSE Linux Enterprise, delivered when they are released to SUSE Linux Enterprise customers.

SUSE Linux Enterprise Server for IBM Power is an enterprise-grade Linux distribution that is optimized for IBM Power processor-based systems. It is designed to deliver enhanced reliability and provide a high-performance platform that meets growing business demands. This distribution helps accelerate innovation while improving deployment times.

For more information about SUSE Linux Enterprise Server for IBM Power, see [SUSE Linux Enterprise Server for IBM Power](#).

Debian-based distributions

Debian is a Linux distribution that is known for its stability, reliability, security, and extensive software repositories. It consists entirely of no-charge software and serves as the foundation for many other distributions, most notably Ubuntu, which is also supported on IBM Power servers.

Ubuntu is optimized for workloads in mobile, social, cloud, big data, analytics, and ML environments. With deployment tools such as Juju and MAAS, Ubuntu simplifies the management of these workloads. Starting with Ubuntu 22.04 LTS, support is available for Power9 and Power10 processors.

For more information about Ubuntu Server, see [Ubuntu](#).

For information about how to install Linux on IBM Power10 servers, see [Installing Linux on Power10 systems](#).

Linux support on Power10 servers

Multiple Linux distributions are supported on IBM Power servers. Table 10-1 shows the supported versions of those Linux distributions on IBM Power10 processor-based systems.

Table 10-1 Linux distributions for Power10 processor-based systems

IBM Power10 processor-based systems	PowerVM logical partitions
<ul style="list-style-type: none">▶ 9043-MRX (IBM Power E1050)▶ 9105-22A (IBM Power S1022)▶ 9105-22B (IBM Power S1022s)▶ 9105-41B (IBM Power S1014)▶ 9105-42A (IBM Power S1024)▶ 9786-22H (IBM Power L1022)▶ 9786-42H (IBM Power L1024)	<ul style="list-style-type: none">▶ RHEL 9.0 and any subsequent RHEL 9.x releases▶ RHEL 8.4 and any subsequent RHEL 8.x releases▶ SUSE Linux Enterprise Server 15 SP3 and any subsequent SUSE Linux Enterprise Server 15 updates▶ Red Hat OpenShift Container Platform 4.9 or later▶ Ubuntu 22.04 or later^a
9080-HEX (IBM Power E1080)	<ul style="list-style-type: none">▶ RHEL 9.0 and any subsequent RHEL 9.x releases▶ RHEL 8.4 and any subsequent RHEL 8.x releases▶ RHEL 8.2 (Power9 Compatibility mode only)^b▶ SUSE Linux Enterprise Server 15 SP3 and any subsequent SUSE Linux Enterprise Server 15 updates▶ SUSE Linux Enterprise Server 12 SP5 (Power9 Compatibility mode only)▶ Red Hat OpenShift Container Platform 4.9 or later▶ Ubuntu 22.04 or later^a
9028-21B (IBM Power S1012)	<ul style="list-style-type: none">▶ RHEL 9.2, for PowerLE or later▶ Red Hat OpenShift Container Platform 4.15 or later▶ SUSE Linux Enterprise Server 15 SP6, and any subsequent SUSE Linux Enterprise Server 15 updates▶ Ubuntu 22.04 or later^a

a. Ubuntu on Power support is available directly from Canonical.

b. Red Hat Business Unit approval is required for using RHEL 8.2 on IBM Power10 processor-based systems.

IBM Power10 processor-based systems support the following configurations per logical partition (LPAR):

- ▶ SUSE Linux Enterprise Server 15 SP4: Up to 64 TB of memory and 240 processor cores.
- ▶ SUSE Linux Enterprise Server 15 SP3: Up to 32 TB of memory and 240 processor cores.
- ▶ RHEL 8.6 or later: Up to 64 TB of memory and 240 processor cores.
- ▶ RHEL 8.4 and 9.0: Up to 32 TB of memory and 240 processor cores.
- ▶ SUSE Linux Enterprise Server 12 SP5 and RHEL 8.2: Up to 8 TB of memory and 120 processor cores.

The best practice for a Linux distribution for a specific server is always the latest version that is optimized for that hardware. The distributions that are listed are the OS versions that are supported for the corresponding IBM Power servers. For more information each respective distribution:

- ▶ [SUSE Linux Enterprise Server](#)
- ▶ [RHEL](#)
- ▶ [Ubuntu](#)

Note: CoreOS is supported as a part of Red Hat OpenShift Container Platform. For more information about Red Hat OpenShift Container Platform, see [Getting started with Red Hat OpenShift on IBM Cloud](#) and [Red Hat OpenShift Container Platform](#).

For more information about Linux on Power and various supported combinations, see the [official IBM Documentation](#).

10.1.2 Linux on Power features and functions

Linux on Power includes specific support for features in the IBM Power platform. This section describes some of them.

Hybrid Network Virtualization

Hybrid Network Virtualization (HNV) enhances the availability and performance of Linux partitions on IBM Power servers by using two key technologies: Single Root I/O Virtualization (SR-IOV) and Live Partition Mobility (LPM).

- ▶ SR-IOV

SR-IOV enables a single I/O adapter to be shared simultaneously across multiple LPARs, delivering hardware-level performance without adding extra CPU processing. This efficiency is achieved because the I/O adapter handles virtualization directly at the hardware level.

Before SR-IOV, each network adapter was dedicated to a single partition and required virtualization through the Virtual I/O Server (VIOS) to be shared across multiple partitions. As network speeds increased, managing virtualized network traffic through VIOS demanded more compute resources, introducing extra processing and often increasing network latency.

With SR-IOV, network adapters are simplified, resulting in improved performance and reduced VIOS processing. However, this configuration limits the ability of partitions to benefit from LPM.

- ▶ LPM

The IBM PowerVM LPM feature enables the seamless relocation of a running virtual partitions from one physical IBM Power server to another one. This capability is especially valuable for avoiding service disruptions during planned events such as maintenance or upgrades, allowing for complete server evacuation without interrupting workloads.

In addition to supporting planned downtime, LPM facilitates dynamic workload balancing, helping optimize resource usage and performance across the infrastructure.

To enable LPM, virtual partitions must use virtualized I/O adapters. However, when partitions use SR-IOV adapters directly, they cannot use LPM. To address this limitation, IBM developed Hybrid Network Virtualization (HNV), which virtualizes SR-IOV resources. This approach enables partitions to benefit from the high performance of SR-IOV while still supporting LPM functions.

In summary, HNV on IBM Power servers combines the high-performance I/O capabilities of SR-IOV with the flexibility of LPM. This integration enables improved performance, simplified network configuration, and seamless workload migration.

IBM Power firmware level FW950.00 and Hardware Management Console (HMC) 9.2.950.0 with compatible Linux distributions introduced support for HNV on Linux. This advancement enables LPARs running Linux to benefit from the efficiency and performance of SR-IOV logical ports (LPs) while participating in key mobility operations. These operations include both active and inactive LPM and Simplified Remote Restart (SRR).

Enabling HNV is a straightforward process during the configuration of an SR-IOV LP, where a new **Migratable** option can be selected.

Behind the scenes, HNV uses Linux active-backup bonding. This bonding mechanism is essential for enabling LPM for virtual partitions that are configured to use SR-IOV LPs, helping ensure a seamless migration process.

Requirements for HNV

The following requirements and conditions must be met to perform the HNV operation:

- ▶ HMC Version 9 Release 2 Maintenance Level 950 or later
- ▶ VIOS 3.1.2.0 or later
- ▶ Power Hypervisor with firmware at level FW950 or later
- ▶ powerpc-utils 1.3.8 or later for RHEL 8.4+ and SUSE Linux Enterprise Server15 SP3
- ▶ powerpc-utils 1.3.10 or later for SUSE Linux Enterprise Server15 SP4+
- ▶ Back-end virtual device support:
 - IBM virtual Ethernet device (ibmveth)
 - SUSE Linux Enterprise Server15 SP3 or later
 - RHEL 8.4 or later
 - RHEL 9.0 or later
 - IBM virtual network interface (ibmvNIC)
 - RHEL 8.6 or later
 - RHEL 9.0 or later
- ▶ DynamicRM-2.0.7-7.ppc64le.rpm
- ▶ Bonding module

For more information about HNV, see [Hybrid Network Virtualization - Using SR-IOV for Optimal Performance and Mobility](#).

Guest secure boot with static keys

IT security is essential in the digital era. Because businesses increasingly depend on technology, protecting sensitive data and preventing cyberattacks becomes a critical priority. Guest secure boot with static keys was recently introduced in Linux on IBM Power. This section provides a brief technical overview of this function.

Secure boot and verified boot are firmware and software mechanisms that protect the integrity of OS boot components. Malicious software, such as boot kits and root kits, can subvert the bootloader and OS security during the boot process. These threats are countered by public key cryptography. The binary hash of boot components is signed with private keys, and the signatures are verified by using the corresponding public keys. The boot component image hash is compared with the signature hash, and the boot process continues only if they match. This verification procedure helps ensure that the integrity of the boot components remains intact.

Linux LPAR secure boot helps ensure the integrity of the Linux boot stack. The hypervisor and partition firmware form the core root of trust. The partition firmware verifies the appended signature on the GRUB image before transferring control to GRUB. Similarly, GRUB verifies the appended signature on the kernel image before starting the OS. This process helps ensure that every image that runs during boot is verified and trusted.

By default, Linux LPAR secure boot uses static key management. In this model, each image embeds the keys that are required to verify the next image in the boot sequence. For example, the keys that are used to verify the GRUB image are embedded in the firmware image. Similarly, the keys that are used to verify the kernel image are embedded in the GRUB image. These keys are predefined and cannot be modified at run time. Any changes to the static keys require both firmware and OS updates.

Figure 10-2 shows how the static, key-based guest secure boot solution works. The numbers represent the chronological order of operation for each individual boot component.

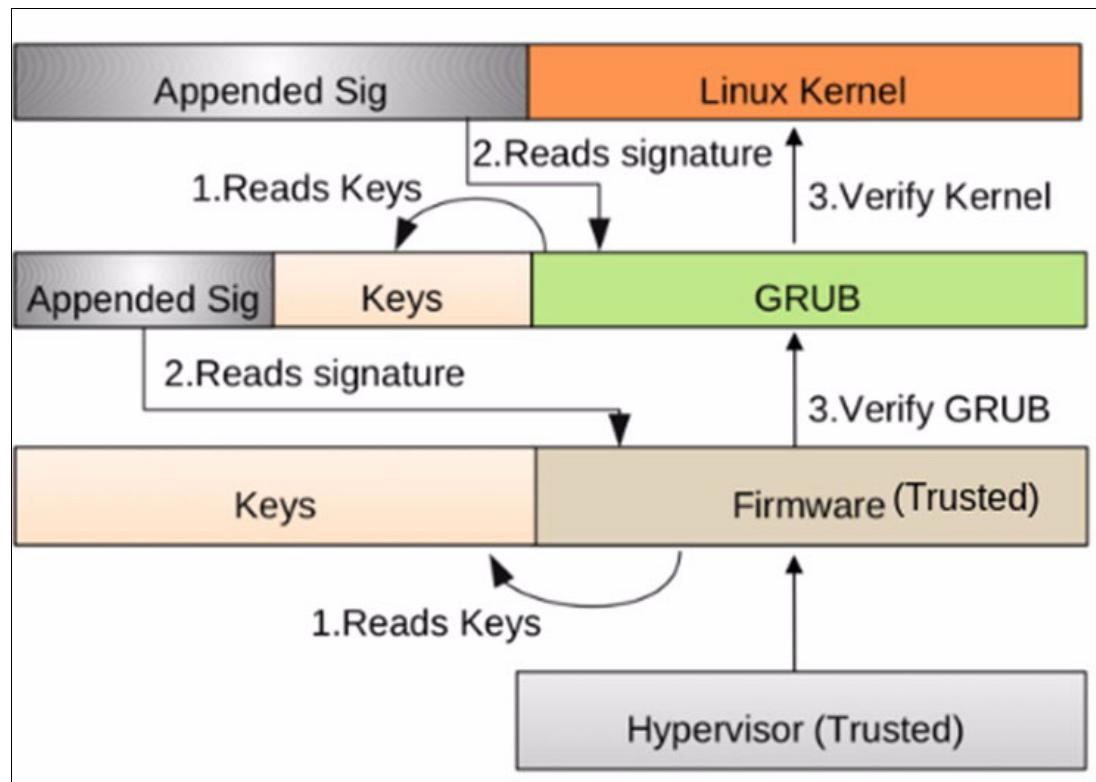


Figure 10-2 Static key based secure boot solution

Secure boot modes

The HMC provides three secure boot modes:

- ▶ Disabled
- ▶ Enabled and log only
- ▶ Enabled and enforced

Linux on Power supports two of these three modes:

- ▶ Disabled
- ▶ Enabled and enforced

Administrators can configure this setting from the HMC for each LPAR. The default setting is Disabled. This setting is available under Advanced Settings.

For more information, including supported releases and features, see [Guest secure boot with static keys](#).

Kernel-based Virtual Machine in a PowerVM LPAR

Kernel-based Virtual Machine (KVM) provides an another virtualization option on Power10 servers that run PowerVM. KVM brings the performance, flexibility, and capabilities of KVM virtualization technology to a PowerVM LPAR. An LPAR running a KVM-enabled Linux distribution can host PPC64-LE KVM guests. These guests use the existing resources that are assigned to the LPAR.

KVM in a PowerVM LPAR uses the industry-standard Linux KVM virtualization stack and integrates into an existing Linux virtualization ecosystem.

For more information about KVM in Power10, see “Kernel-based Virtual Machine” on page 39.

10.2 Containerization solutions on IBM Power

Modernizing an IBM Power infrastructure with containers is a strategic approach for organizations seeking to enhance agility, efficiency, and scalability while using their existing investment in Power servers. Containerization enables applications and their dependencies to be packaged together, allowing for faster deployment cycles, improved resource usage, and a consistent experience across different environments.

Containerization solutions on IBM Power servers offer a powerful and flexible method for deploying and managing applications by using the performance and reliability of the Power Architecture. This section describes the containerization technologies that are available on IBM Power.

10.2.1 Red Hat OpenShift

Red Hat OpenShift is an enterprise-grade container orchestration platform that is developed by Red Hat. It is built on Kubernetes, which is an open-source system for automating the deployment, scaling, and management of containerized applications. Red Hat OpenShift simplifies container management by providing an integrated environment that includes container orchestration, continuous integration and continuous deployment or delivery (CI/CD), monitoring, and security. It enables organizations to build, deploy, and scale applications efficiently across hybrid and multi-cloud environments.

One of the key features of Red Hat OpenShift is its developer-centric toolset, which streamlines the application lifecycle. Developers can quickly create, test, and deploy applications by using predefined templates, automated workflows, and integrated development environments (IDEs). Red Hat OpenShift includes built-in support for several programming languages and frameworks, along with Kubernetes-native resources such as pods, services, and persistent storage. These capabilities enable developers to focus on building applications rather than managing infrastructure.

In addition to its developer-focused features, Red Hat OpenShift provides enterprise-level security and management tools to help ensure that containerized applications are secure, scalable, and maintainable. It includes integrated role-based access control (RBAC), automated security updates, and centralized logging and monitoring. Red Hat OpenShift is designed to run in diverse environments, including on-premises data centers and public clouds, making it a suitable solution for organizations seeking to modernize their IT infrastructure and adopt containerization and microservices architecture.

Red Hat OpenShift on IBM Power

Red Hat OpenShift on IBM Power integrates the capabilities of the IBM Power Architecture with a leading Kubernetes-based hybrid cloud application platform. This combination provides a powerful and adaptable environment for modernizing applications and deploying cloud-native workloads. Red Hat OpenShift offers comprehensive features for development, deployment, and management, and IBM Power delivers performance, reliability, scalability, and security, which are beneficial for demanding workloads such as artificial intelligence (AI), big data, and mission-critical systems.

Supporting hybrid cloud deployments, Red Hat OpenShift on IBM Power enables workload flexibility across on-premises and cloud environments. It facilitates application modernization through containerization and microservices, and it is managed efficiently through the Operator Framework and a growing catalog of certified Operators. Key benefits include enhanced performance and scalability, improved reliability, optimized resource usage, robust security, simplified management, potential cost efficiencies, and the ability to use existing Power infrastructures. IBM Cloud Pak further extends the platform's capabilities.

Deployment options include user-provisioned infrastructure (UPI) for greater control and installer-provisioned infrastructure (IPI), primarily on IBM Cloud Power Virtual Server (PowerVS). Deployment considerations include operating system support, hardware requirements, networking, storage, and licensing. Red Hat OpenShift on IBM Power provides a strategic foundation for organizations seeking a high-performance, reliable, and secure platform for modern application initiatives.

IBM Cloud Pak

IBM Cloud Pak is AI-powered software for hybrid cloud environments that is designed to support digital transformation through capabilities in prediction, security, automation, and modernization. It enables organizations to develop applications and deploy them anywhere, integrate security across the IT landscape, and automate operations by using intelligent workflows. IBM Cloud Pak can be deployed across any cloud to accelerate development, streamline integration, and enhance collaboration and efficiency.

IBM Cloud Pak is designed to help you in the following areas:

- ▶ Modernize: Develop and use cloud services anywhere, from any cloud.
- ▶ Predict outcomes: Collect, organize, and analyze data regardless of its type or where it lives.
- ▶ Automate at scale: Implement intelligent workflows in your business by using AI-powered automation.
- ▶ Protect your business: Generate deeper insights into threats and risks across hybrid multicloud environments.

Figure 10-3 shows IBM Cloud Pak on Power.

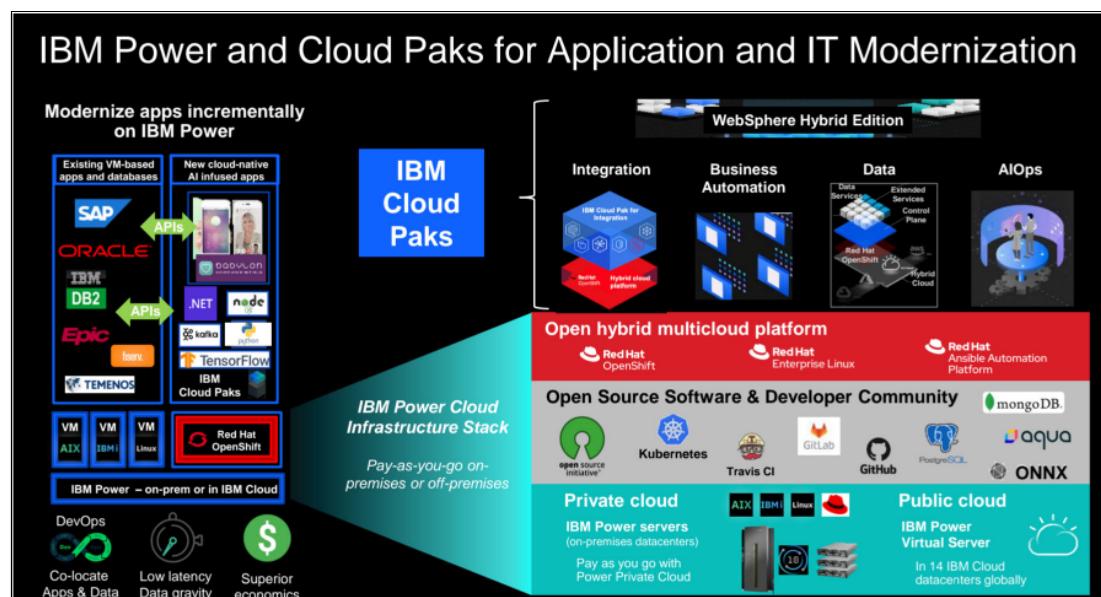


Figure 10-3 IBM Cloud Pak on IBM Power

More businesses are recognizing that AI is essential for digital transformation and gaining a competitive advantage. To scale AI across the enterprise, organizations must build an information architecture that connects multiple data sources, helps ensure data quality, and supports AI data requirements. As the value of AI continues to grow, addressing these challenges becomes a priority for many organizations.

IBM Cloud Pak is pre-integrated, containerized software that is built on Red Hat OpenShift. It is designed to help organizations develop and use cloud services from any location and on any cloud, enabling seamless modernization and data usage. IBM Cloud Pak enables flexible and rapid deployment management through a governed, protected, and unified platform that delivers consistency across software tools. It is continuously available from the data center to the edge.

10.2.2 Kubernetes

Kubernetes is an open-source container orchestration platform that is used to schedule and automate the deployment, management, and scaling of containerized applications. It enables organizations to deliver a highly productive hybrid multicloud computing environment for performing complex infrastructure and operations tasks. Kubernetes also supports cloud-native development by enabling a *build-once, deploy-anywhere* approach. It automates container-related tasks throughout the application lifecycle, including deployment, provisioning, management, and maintenance, as shown in Figure 10-4.

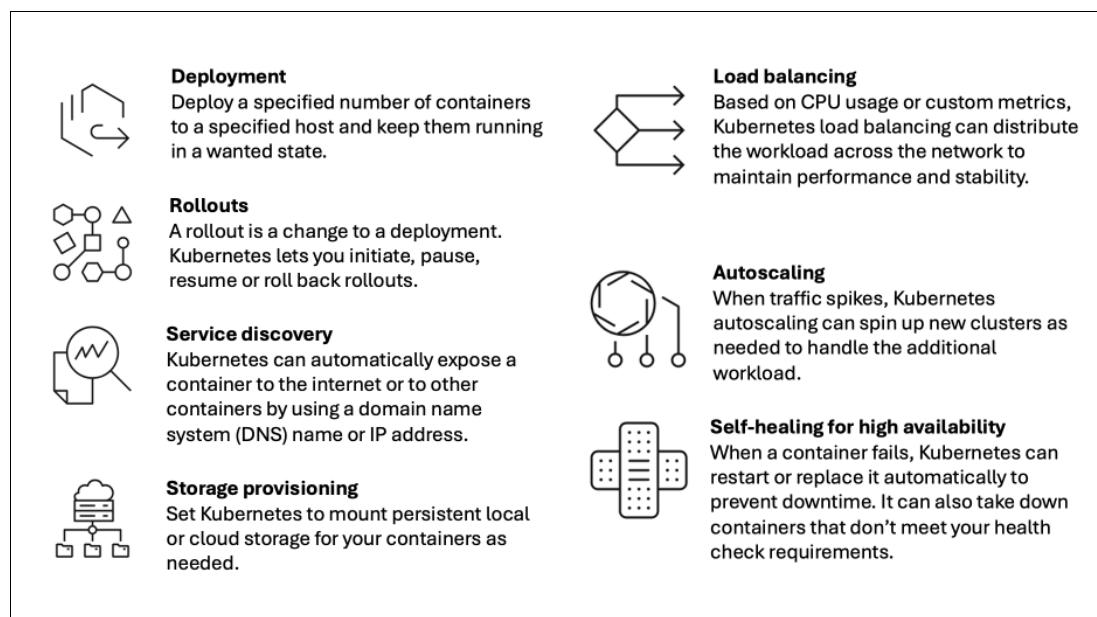


Figure 10-4 Application lifecycle management

Kubernetes architecture and components

Kubernetes architecture consists of two main parts: the control plane components and the components that manage individual nodes, as shown in Figure 10-5.

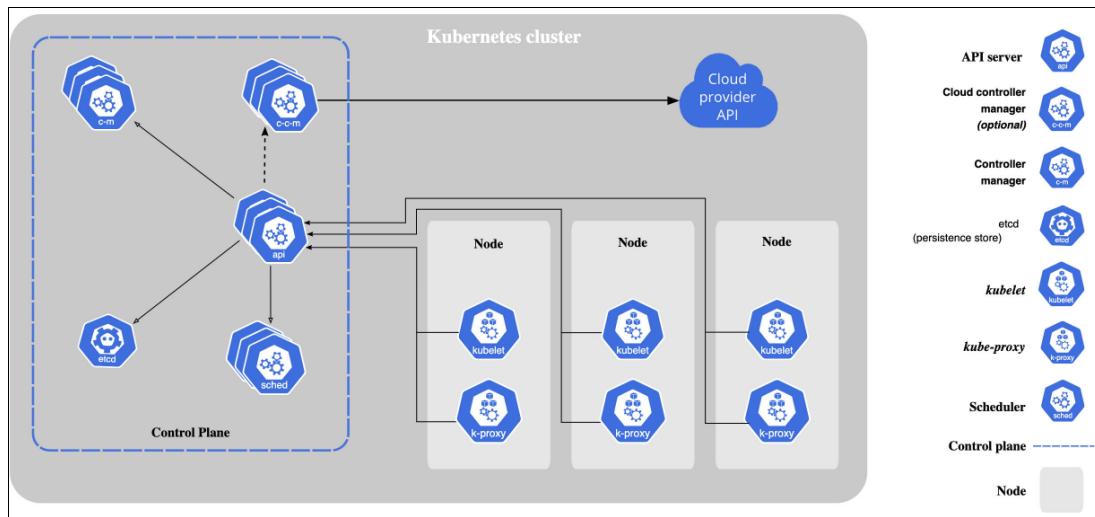


Figure 10-5 Kubernetes components

A node consists of pods, which are groups of containers that share computing resources and a network. Pods are the unit of scalability in Kubernetes. If a container within a pod receives more traffic than it can handle, Kubernetes replicates the pod to other nodes in the cluster. The control plane automatically schedules pods across the nodes in the cluster.

Each Kubernetes cluster includes a master node that manages the control plane. The master node runs a scheduler service that automates when and where containers are deployed, based on developer-defined deployment requirements and available computing capacity.

The main components in a Kubernetes cluster are the kube-apiserver, etcd, kube-scheduler, kube-controller-manager, and cloud-controller-manager, which are described here:

- ▶ Application programming interface (API) server

The apiserver in Kubernetes exposes the Kubernetes API, which is the interface that is used to manage, create, and configure Kubernetes clusters. It serves as the entry point for all commands and queries.

- ▶ etcd

etcd is an open-source distributed key-value store that is used to manage the critical information that is required by distributed systems. In Kubernetes, etcd stores configuration data, state data, and metadata.

- ▶ Scheduler

The scheduler tracks newly created pods and selects nodes for them to run on. It considers resource availability, allocation constraints, hardware and software requirements, and other factors.

- ▶ Controller-manager

The Kubernetes controller manager runs a set of built-in controllers that operate in a control loop. These controllers monitor the shared state of the cluster and communicate with the apiserver to manage resources, pods, and service endpoints. The controller manager includes separate controller processes that are bundled together to reduce complexity and run as a single process.

- ▶ Cloud-controller-manager

This component functions similarly to the controller manager. It connects to a cloud provider's API and separates components that interact with the cloud platform from ones that operate solely within the cluster.

Node components

Node components run on every node, maintaining running pods and providing the Kubernetes runtime environment:

- ▶ kubelet: Helps ensure that Pods are running, including their containers.
- ▶ kube-proxy (optional): Maintains network rules on nodes to implement services.
- ▶ Container runtime: Software that is responsible for running containers.

Kubernetes use cases

Enterprise organizations use Kubernetes to support the following use cases, which all play a crucial role in making up the modern IT infrastructure:

- ▶ Microservices architecture or cloud-native development

Kubernetes helps ensure that each microservice has the resources that it needs to run effectively while minimizing the operational processing of manually managing multiple containers.

- ▶ Hybrid multicloud environments

Hybrid cloud combines and unifies public cloud, private cloud, and on-premises data center infrastructures to create a single, flexible, and cost-optimized IT environment. Today, hybrid cloud merges with multicloud (public cloud services from multiple vendors) to form a hybrid multicloud architecture. This approach increases flexibility and reduces dependency on a single vendor, helping organizations avoid vendor lock-in.

Because Kubernetes provides the foundation for cloud-native development, it plays a key role in hybrid multicloud adoption.

- ▶ Applications at scale

Kubernetes supports large-scale cloud application deployment with autoscaling. This process enables applications to scale up or down automatically in response to changes in demand, helping ensure speed, efficiency, and minimal downtime. The elastic scalability of Kubernetes enables resources to be added or removed based on fluctuations in user traffic, such as during flash sales on retail websites.

- ▶ Application modernization

Kubernetes provides the modern cloud platform that is needed to support application modernization by migrating and transforming monolithic applications into cloud-native applications that are built on a microservices architecture.

- ▶ DevOps practices

Automation is central to DevOps, which accelerates the delivery of high-quality software by combining and automating the work of software development and IT operations (ITOps) teams. Kubernetes supports DevOps teams by automating the configuration and deployment of applications, enabling rapid application development and updates.

- ▶ AI and ML

ML models and large language models (LLMs) that support AI include components that are difficult and time-consuming to manage individually. By automating configuration, deployment, and scalability across cloud environments, Kubernetes provides the agility and flexibility that are needed to train, test, and deploy these complex models.

For more information about Kubernetes, see [What is Kubernetes?](#)

10.2.3 Docker

Docker is an open-source platform that enables developers to build, deploy, run, update, and manage containers. It is a widely used containerization tool. Docker has become so ubiquitous that the terms “Docker” and “containers” are often used interchangeably.

Containers simplify the development and delivery of distributed applications. They become more prominent as organizations adopt cloud-native development and hybrid multicloud environments. Although developers can create containers without Docker by using capabilities that are built in to Linux and other OSs, Docker streamlines and accelerates the containerization process. Like other containerization technologies such as Kubernetes, Docker plays a critical role in modern software development, particularly in microservices architecture.

Docker architecture

Docker uses a client/server architecture. Here is a breakdown of the core components that are associated with Docker, along with other Docker terms and tools:

- ▶ Docker host

A Docker host is a physical or virtual machine (VM) running Linux (or another Docker Engine-compatible OS).

- ▶ Docker Engine

Docker Engine is a client/server application that consists of the Docker daemon, a Docker API that interacts with the daemon, and a command-line interface (CLI) that talks to the daemon.

- ▶ Docker daemon

The Docker daemon is a service that creates and manages Docker images by using commands that are received from the client. It serves as the control center for Docker operations.

- ▶ Docker client

The Docker client provides a CLI that uses the Docker API (a REST API) to communicate with the Docker daemon over UNIX sockets or a network interface.

- ▶ Docker objects

Docker objects are components of a Docker deployment that package and distribute applications. These objects include images, containers, networks, volumes, plug-ins, and other resources.

Docker containers are live, running instances of Docker images. Although Docker images are read-only files, containers are ephemeral, executable environments. Users can interact with containers, and administrators can modify their settings and runtime conditions by using Docker commands.

- ▶ Docker images

Docker images contain executable application source code along with all the tools, libraries, and dependencies that are required to run the application as a container. When a developer runs a Docker image, it becomes a container instance.

Docker images consist of multiple layers, with each layer representing a version of the image. When a developer modifies an image, Docker creates a top layer that replaces the previous one as the current version. Earlier layers are retained for rollback purposes or reuse in other projects.

Each time that a container is created from a Docker image, Docker adds a layer that is named the *container layer*. Changes that are made to the container, such as adding or deleting files, are saved to this layer. These changes persist only while the container is running.

This layered, iterative image-creation process improves efficiency because multiple live container instances can run from a single base image. These instances share a common underlying stack.

- ▶ Docker build

Docker build is a command that has tools and features for building Docker images.

- ▶ Dockerfile

Every Docker container begins with a simple text file that is named a *Dockerfile*, which contains instructions for building the Docker image. The Dockerfile automates the image creation process by providing a list of CLI instructions that Docker Engine runs to assemble the image.

Although the list of Docker commands is extensive, it is standardized. Docker operations function consistently, regardless of the image contents, infrastructure, or environment variables.

- ▶ Docker Hub

Docker Hub is a public repository for Docker images and describes itself as the world's largest library and community for container images. It includes images that are published by Docker, Inc., certified images from the Docker Trusted Registry, and thousands of other community-contributed images.

- ▶ Docker Desktop

Docker Desktop is an application for macOS and Windows that includes Docker Engine, the Docker CLI, Docker Compose, Kubernetes, and other components. It also provides integrated access to Docker Hub.

- ▶ Docker registry

A Docker registry is a scalable, open-source system for storing and distributing Docker images. It enables developers to track image versions in repositories by using tags for identification. This tracking and identification are accomplished by using Git, a version control tool.

Docker use cases

From cloud migration to CI/CD to AI/ML, Docker offers several business-critical use cases for organization in their modernization journey.

Figure 10-6 captures various Docker use cases that are available for businesses.

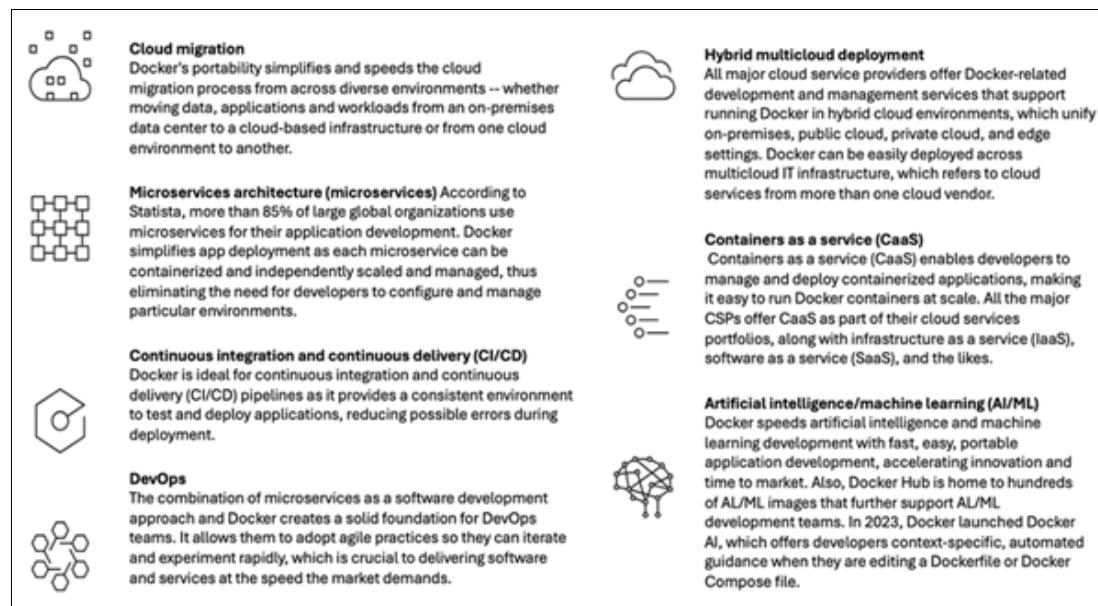


Figure 10-6 Docker use cases

In 2023, Docker introduced Docker AI², its first AI-powered product that is designed to enhance developer productivity. Docker AI provides context-specific, automated guidance by drawing on the collective expertise of the Docker developer community.

Docker AI helps developers while they edit Dockerfiles or Docker Compose files, debug local docker build operations, or run local tests. It automatically generates best practices and recommends secure, up-to-date images for applications. By reducing the time that is spent on tools and infrastructure, Docker AI enables developers to focus more on application development.

10.2.4 Podman

Podman is an open-source tool for developing, managing, and running containers. Developed by Red Hat engineers in collaboration with the open-source community, Podman manages the entire container ecosystem by using the libpod library.

Podman features a daemonless and inclusive architecture, making it a secure and accessible option for container management. Its companion tools, such as Buildah and Skopeo, enable developers to customize container environments to meet specific requirements. Developers can also use Podman Desktop, a GUI for managing containers in local environments. Podman runs on various Linux distributions, including RHEL, Fedora, CentOS, and Ubuntu.

² Source:

<https://www.docker.com/press-release/announces-ai-boosting-developer-productivity-through-automated-guidance/#>

Podman distinguishes itself from other container engines through its daemonless architecture, which eliminates the need for a root-privileged process to run containers. Traditional container engines rely on background daemons to manage containers without a user interface (UI). Podman removes this dependency, enabling users to run containers without interacting with a root-owned daemon. It also supports rootless containers, enabling users to create, run, and manage containers without requiring administrative privileges. This approach enhances accessibility and reduces security risks.

Also, Podman starts each container with a Security-Enhanced Linux (SELinux) label, giving administrators greater control over the resources and capabilities that are available to container processes.

Users can start Podman from the CLI to pull container images from a repository and run them. Podman uses the configured container run time to create and manage running containers. Unlike traditional container engines that rely on a dedicated daemon, Podman integrates with systemd (a system and service manager for Linux operating systems) to manage updates and keep containers running in the background.

By integrating Podman with systemd, users can generate control units for containers and run them with systemd automatically enabled.

Podman also provides a RESTful API for managing containers. A REST API conforms to the constraints of the REST architectural style and enables interaction with RESTful web services. Using the Podman REST API, developers can issue commands from platforms such as cURL, Postman, Google's Advanced REST Client, and other similar tools.

Podman delivers high-performance container management capabilities that are comparable to ones of leading container engines while offering enhanced flexibility, accessibility, and security features that meet the needs of modern development teams.

Podman can help you in the following ways:

- ▶ Manage container images and the full container lifecycle, including running, networking, checkpointing, and removing containers.
- ▶ Run and isolate resources for rootless containers and pods.
- ▶ Supports OCI and Docker images and a Docker-compatible CLI.
- ▶ Create a daemonless environment to improve security and reduce idle resource consumption.
- ▶ Deploy a REST API to support Podman's advanced functions.
- ▶ Implement a checkpoint/restore function for Linux containers with Checkpoint/Restore in Userspace (CRIU). CRIU can freeze a running container and save its memory contents and state to disk so that containerized workloads can be restarted faster.
- ▶ Automatically update containers. Podman detects whether an updated container fails to start and automatically rolls back to the last working version. This approach provides new levels of reliability for applications.

For more information about Podman, see [What is Podman?](#)



Part 3

Appendices

This part provides more details to support the modernization of your IBM Power infrastructure and applications.

Appendix A, “Components that are used in modernization on IBM Power” on page 363 lists components and tools that can support the modernization of an IBM Power infrastructure. This list is intended to illustrate the range of available options for building a modern infrastructure and updating applications.

Appendix B, “Modernization by using cloud-native tools” on page 407 provides guidance about using cloud-native tools within Skytap on Azure. It highlights key Azure-native services that support integration, scalability, and automation for IBM i, AIX, and Linux on IBM Power environments.

Appendix C, “Details for performance claims” on page 429 provides supporting information for the performance and financial claims that are presented in this publication.

The following appendixes are included in this part:

- ▶ Appendix A, “Components that are used in modernization on IBM Power” on page 363
- ▶ Appendix B, “Modernization by using cloud-native tools” on page 407
- ▶ Appendix C, “Details for performance claims” on page 429



A

Components that are used in modernization on IBM Power

This publication describes various approaches for modernizing applications and workloads on the robust IBM Power platform, ultimately increasing the value that is delivered to your business. Although IBM Power is widely recognized for its reliability and high performance in running mission-critical applications, this publication demonstrates how integrating modern digital interfaces and artificial intelligence (AI) can further enhance its capabilities and generate business advantages.

This appendix outlines several components that support the modernization of IBM Power infrastructure. Although not exhaustive, the appendix presents a selection of tools that are organized into broad categories to help guide modernization efforts.

The following topics are described in this appendix:

- ▶ “Red Hat” on page 364
- ▶ “IBM” on page 379
- ▶ “Open-source solutions” on page 384
- ▶ “Independent software vendors” on page 396

Red Hat

Red Hat plays a pivotal role in modernizing IT infrastructure and applications by helping organizations migration to cloud-native environments. Through Red Hat OpenShift, businesses can replatform existing workloads onto a Kubernetes-based platform and decompose monolithic applications into microservices. This transformation enhances performance, scalability, security, and compliance.

Red Hat's modernization strategy also includes the adoption of DevOps practices, continuous integration and continuous deployment or delivery (CI/CD), and site reliability engineering (SRE), all of which streamline workflows and improve engineering productivity.

Modernizing with Red Hat delivers cost savings and operational efficiencies. By containerizing applications and using Red Hat OpenShift's self-healing and auto-scaling capabilities, organizations can optimize resource usage and reduce maintenance costs. Red Hat's tools and frameworks also support hybrid and multicloud deployments, providing flexibility and helping ensure seamless application performance across diverse environments.

This comprehensive approach to modernization helps businesses remain competitive, improve user experiences, and achieve their digital transformation goals.

Red Hat OpenShift

Red Hat OpenShift provides a robust and scalable platform for developing, deploying, and managing containerized applications. It offers a consistent hybrid cloud foundation, enabling seamless operations across on-premises, public cloud, and edge environments. Red Hat OpenShift enhances developer productivity through integrated developer tools, CI/CD pipelines, and automated workflows.

Its enterprise-grade security features, which include built-in compliance and automated updates, help ensure that applications remain secure and current. Also, support for Multi-Architecture Clusters and Single Node Red Hat OpenShift enables organizations to optimize infrastructure for diverse workloads and resource-constrained environments, making Red Hat OpenShift a versatile solution for modern IT requirements.

Recent support

Red Hat OpenShift recently introduced two key features: Single Node Red Hat OpenShift and Multi-Architecture Clusters. Both features provide advantages in the IBM Power environment. This section describes these advantages.

Single Node Red Hat OpenShift

A standard Red Hat OpenShift implementation typically uses multiple control plane nodes and worker nodes to help ensure high availability (HA). For environments that do not require hardware redundancy and aim to conserve resources, Single Node Red Hat OpenShift is an effective option. This specialized deployment consolidates control plane and worker node functions on a single server. It is useful for edge computing environments where space, power, and connectivity are limited. Designed for autonomous operation, Single Node Red Hat OpenShift is ideal for scenarios such as intermittent connectivity, portable clouds, and 5G radio access networks (RANs) near base stations. By using a single node, organizations can deploy a full Red Hat OpenShift environment with fewer resources, simplifying the setup and management of Kubernetes clusters in constrained environments.

The benefits of Single Node Red Hat OpenShift include reduced resource requirements and simplified management. By combining control plane and worker node capabilities, it provides a consistent Red Hat OpenShift experience across deployment sizes, from large data centers to remote edge locations. This consistency enables organizations to use the same tools and skills across their infrastructure, helping ensure seamless operations and upgrades. Its ability to run autonomously makes it suitable for environments with limited or unreliable connectivity, enabling continuous operations in challenging conditions. Overall, Single Node Red Hat OpenShift offers a flexible and efficient solution for deploying Kubernetes in resource-constrained environments.

Multiple architecture clusters

Red Hat OpenShift 4.14 introduced the Multi-Architecture Compute feature to the Red Hat OpenShift Container Platform on IBM Power. This feature enables a single heterogeneous cluster, enabling fit-for-purpose computing so that clients can align tasks and applications to CPU strengths and software availability rather than a single architecture. Red Hat OpenShift 4.15 expanded this support by enabling clusters with an IBM Power control plane and x86 architecture worker nodes.

With Multi-Architecture Compute, you may use two compute architectures, such as ppc64le and amd64, within a single cluster. This feature provides new opportunities for versatility and optimization in composite solutions that span multiple architectures.

For more information about multiple architecture clusters on IBM Power, see *Creating Red Hat OpenShift Multiple Architecture Clusters with IBM Power*, SG24-8565.

Red Hat OpenShift add-ons

Red Hat OpenShift offers various add-ons that enhance cluster functions and capabilities. These add-ons include services such as monitoring, logging, security, and networking, which you can integrate by using the Red Hat OpenShift Cluster Manager. For example, you can add the cluster-logging-operator to enable comprehensive logging or integrate external monitoring tools such as Amazon CloudWatch. These add-ons streamline operations, improve visibility, and help optimize performance and reliability. By using these add-ons, organizations can tailor their Red Hat OpenShift deployments to meet specific requirements and achieve greater efficiency and scalability.

Pipelines and Tekton

Red Hat OpenShift Pipelines is a cloud-native continuous integration and continuous delivery (CI/CD) solution that is based on the Tekton framework. It enables developers to create advanced CI workflows for their applications, automating and accelerating the delivery process. Each step in the CI/CD pipeline runs in its own container, allowing independent scaling to meet pipeline demands. This serverless approach removes the need to manage a central CI/CD server and provides full control over delivery pipelines, plug-ins, and access control. Red Hat OpenShift Pipelines integrates with the Red Hat OpenShift console, enabling developers to configure and run pipelines directly alongside their applications.

Tekton, the framework underlying Red Hat OpenShift Pipelines, is an open-source project that provides Kubernetes-native CI/CD components. Its flexible architecture supports standard Kubernetes-native pipeline definitions, extensibility for building images with various Kubernetes tools, and portability across any Kubernetes distribution. Tekton also includes a robust command-line interface (CLI) for interacting with pipelines and an integrated experience in the Red Hat OpenShift web console. By using Tekton, Red Hat OpenShift Pipelines delivers a streamlined and scalable CI/CD solution that helps organizations automate application delivery, reduce time to market, and apply DevSecOps practices to identify and fix vulnerabilities early in the development process.

Argo CD

Argo CD is a declarative GitOps continuous delivery tool that is designed for Kubernetes. It automates application deployment and lifecycle management by using Git repositories as the source of truth for the wanted application state. This approach helps ensure that application definitions, configurations, and environments are version-controlled and auditable. Argo CD continuously monitors running applications, comparing the live state with the wanted state in the Git repository. If discrepancies occur, Argo CD can automatically or manually synchronize the live state to match the wanted state, helping ensure consistency and reliability.

One key feature of Argo CD is its support for multiple configuration management and templating tools, such as Kustomize, Helm, and Jsonnet. It also provides robust security features, including single sign-on (SSO) integration with various identity providers and role-based access control (RBAC) for authorization. In addition, Argo CD supports multi-tenancy and can manage deployments across multiple clusters, making it a versatile tool for complex, large-scale Kubernetes environments. By automating application deployments and providing a clear, auditable process, Argo CD helps organizations deliver software faster and more reliably.

Red Hat OpenShift Serverless

Red Hat OpenShift Serverless simplifies the development and deployment of cloud-native applications by abstracting infrastructure complexities. Based on the open-source Knative project, it enables developers to focus on writing code without managing servers, scaling, or resource provisioning. This approach improves productivity by automatically scaling applications up or down based on demand, including scaling to zero when there are no active requests. It helps ensure efficient resource usage and cost savings because you pay only for the resources that you use.

Red Hat OpenShift Serverless integrates with other Red Hat OpenShift services, such as Service Mesh, Pipelines, and GitOps to provide a comprehensive development and deployment experience. It supports hybrid and multi-cloud environments, helping ensure that applications remain portable and consistent across platforms. Red Hat OpenShift Serverless also includes built-in security features, such as integrated security monitoring and automation to help ensure that applications are secure, reliable, and highly available. By using Red Hat OpenShift Serverless, organizations can accelerate the time to market for new services and features while optimizing resource usage and reducing operational costs.

Odo

Odo is an open-source, developer-focused tool that simplifies application development for Kubernetes and Red Hat OpenShift. It provides a fast, iterative development experience by enabling developers to build, test, and deploy applications directly from their local machine to a cluster. Odo abstracts the complexities of Kubernetes, enabling developers to focus on writing code without managing the underlying infrastructure.

Odo offers several key features that simplify and enhance development for Kubernetes and Red Hat OpenShift environments:

- ▶ **Fast and iterative development:** Enables developers to quickly build, test, and deploy applications directly from their local machine to a Kubernetes cluster, streamlining the workflow.
- ▶ **Language and framework support:** Supports multiple programming languages and frameworks for flexibility.
- ▶ **Integrated development environments (IDE) integration:** Integrates with several IDEs to improve the developer experience.

- ▶ Simple command-line interface (CLI): Provides an intuitive CLI for managing application components and services.
- ▶ Deployment flexibility: Supports on-premises, cloud, and hybrid environments for versatile deployment options.

These features make Odo a powerful tool for developers who must efficiently manage applications on Kubernetes and Red Hat OpenShift.

Red Hat OpenShift Dev Spaces

Red Hat OpenShift Dev Spaces is a Kubernetes-native development solution that provides cloud-based environments for enterprise teams. Built on the open-source Eclipse Che project, it uses Kubernetes and containers to deliver consistent, secure, and zero-configuration development environments. These environments are accessible through a web browser, enabling developers to code, build, test, and run applications on Red Hat OpenShift without managing Kubernetes complexities.

Key features of Red Hat OpenShift Dev Spaces include integration with IDEs such as Visual Studio Code (VS Code) and JetBrains IntelliJ IDEA, support for defining development environments as code by using the devfile format, and robust security measures such as Red Hat OpenShift Open Authorization (OAuth) and integration with lightweight directory access protocol (LDAP) or Active Directory. This tool streamlines the development workflow by providing a fast, familiar experience that is similar to a local IDE, with the added benefits of cloud-based scalability and centralized management.

Cert-manager Operator

The cert-manager Operator is a Kubernetes-native tool that automates certificate management within a cluster. It integrates with Red Hat OpenShift to provide cluster-wide certificate lifecycle management. The cert-manager Operator supports integration with external certificate authorities and offers capabilities for certificate provisioning, renewal, and retirement. By introducing certificate authorities and certificates as resource types in the Kubernetes API, cert-manager enables developers to request and manage certificates on demand.

Key features of the cert-manager Operator include support for various issuer types, such as ACME, CA, self-signed, Vault, Venafi, Nokia NetGuard Certificate Manager (NCM), and Google Cloud Certificate Authority Service (Google CAS). It also provides tools for automatic certificate renewal and self-service certificate management. This operator simplifies the process of securing applications and services within a Kubernetes cluster, helping ensure that certificates remain up to date and properly managed.

Cost Management Metrics Operator

The Cost Management Metrics Operator is a tool that collects and analyzes data from Red Hat OpenShift Container Platform to help organizations manage and optimize costs. It integrates with Red Hat OpenShift to provide detailed insights into resource usage, enabling better forecasting and capacity planning. The operator gathers metrics such as CPU and memory usage, requests, and limits, and can store up to 90 days of historical data. This historical data helps identify trends and patterns, allowing organizations to make informed decisions about cluster capacity and optimize costs.

One key benefit of the Cost Management Metrics Operator is its ability to automate the collection and analysis of cost-related metrics, reducing the manual effort that is required for cost management. By providing a comprehensive view of resource usage and continuously monitoring and optimizing resource allocation, the operator helps organizations achieve cost savings and improve efficiency. It also supports restricted network modes and integrates with various cost management services, making it a versatile tool for different deployment environments. Overall, the Cost Management Metrics Operator enables organizations to better understand and manage IT costs, helping ensure maximum return on their technology investment.

More Red Hat OpenShift products

Red Hat provides more software components and tools that enhance and extend the capabilities of Red Hat OpenShift. These add-ons address specific needs, such as improved security, application management, cloud integration, monitoring, and storage management. They are charged separately. By incorporating these add-ons into an existing Red Hat infrastructure, organizations can streamline operations, improve performance, and use more features to optimize workflows. This section describes some of these add-on products.

Red Hat Quay

Red Hat Quay is a secure and HA container image registry platform. It is a self-managed product that works with most orchestration systems and container environments. Red Hat also offers Quay.io as a hosted software-as-a-service (SaaS) solution.

With the Quay self-managed service, you can run the registry in an offline environment, mirror and replicate repositories, and integrate with Red Hat OpenShift. The fully managed Quay service provides a global content network for serving images, generates vulnerability reports for images, integrates with CI/CD pipelines, and enables building container images from source code commits through GitHub or GitLab.

Red Hat Quay: Self managed

You can install the latest version of Red Hat Quay (3.13) by following the documentation at [Red Hat Quay 3.13 Documentation](#). The operator is available in the OperatorHub on Red Hat OpenShift. After you install the operator, you can use the repository to push and pull images as a self-managed service.

Red Hat Quay: Managed service

The managed service is available at [this repository](#), where you can register and create a repository. After you create the repository, you can push and pull images from it. Pricing is based on the number of private repositories per registry account.

Red Hat Mirror-Registry-EE

A streamlined version of Quay is provided as the mirror registry for Red Hat OpenShift. This lightweight container registry serves as a target for mirroring the required container images of Red Hat OpenShift Container Platform for disconnected installations.

You need a Red Hat OpenShift Container Platform subscription to download the mirror registry installation files. The mirror registry is also available for ppc64le Linux on Power, as shown in Figure A-1 on page 369.



Figure A-1 Downloading the mirror registry installation files

The mirror registry is distributed through a mirror registry Ansible Execution Environment Image (EEI). This image provides the streamlined Quay registry that you can use to upload the required Red Hat OpenShift images for installation.

Note: For more information about how to use the mirror-registry for an offline Red Hat OpenShift installation, see [Creating a mirror registry with mirror registry for Red Hat OpenShift](#).

Red Hat Advanced Cluster Security for Kubernetes

Red Hat Advanced Cluster Security for Kubernetes (RHACS) is a Kubernetes-native security platform that helps organizations secure containerized applications and Kubernetes infrastructure. It secures the software supply chain by integrating with CI/CD pipelines and image registries to provide continuous scanning and assurance of container images, identifying and remediating vulnerabilities early in the development process.

RHACS protects Kubernetes infrastructure by offering Kubernetes Security Posture Management (KSPM) capabilities, hardening the underlying infrastructure, protecting it against targeted exploits, and continuously scanning environments against security benchmarks and best practices. It also defends workloads by providing deploy-time and runtime policies to prevent risky workloads from being deployed or running, and by monitoring system-level events to detect anomalous activity and potential threats.

Key capabilities of RHACS include vulnerability management, compliance auditing, threat detection, network segmentation, and risk profiling. In simple terms, RHACS is a comprehensive security solution that helps organizations build, deploy, and run cloud-native applications more securely.

Red Hat Advanced Cluster Management for Kubernetes

Red Hat Advanced Cluster Management for Kubernetes (RHACM) is a tool that helps organizations manage and govern Kubernetes environments at scale. It provides a centralized platform for managing multiple Kubernetes clusters, whether they are deployed on-premises, in public clouds, or at the edge.

RHACM enables you to manage numerous Kubernetes clusters from a single control plane, simplifying operations and providing a unified view of your Kubernetes infrastructure. It streamlines cluster lifecycle management by facilitating provisioning, upgrading, and decommissioning of clusters. RHACM also enables you to define and enforce policies across clusters to help ensure compliance with security and regulatory requirements, reducing the risk of configuration drift.

In addition, RHACM simplifies application deployment and management across multiple clusters by providing tools for deploying, updating, and rolling back applications. It enforces security policies and checks compliance across all managed clusters, giving you a centralized view of your security posture.

In short, RHACM addresses the challenges of managing Kubernetes in complex, distributed environments. It helps organizations increase operational efficiency, improve security and compliance, and accelerate application delivery.

Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation is a persistent storage and cluster data management solution that integrates with and optimizes Red Hat OpenShift. It offers a distributed, scalable software-defined storage (SDS) platform that provides enterprise-level data management services, enabling applications to interact with data in a consistent, simplified, and scalable way. With multicloud data management capabilities, Red Hat OpenShift Data Foundation enables organizations to extend and federate data across multiple infrastructures.

As a data foundation for modern production workloads and applications, Red Hat OpenShift Data Foundation can run anywhere Red Hat OpenShift is deployed, whether on-premises, in public or private clouds, or at the edge. The platform delivers flexible data access by using standard protocols such as file, block, and object storage, making it suitable for a wide range of workloads. It abstracts the complexities of different underlying storage infrastructures while providing advanced data management services.

Designed for container-based environments, Red Hat OpenShift Data Foundation also supports Red Hat OpenShift Virtualization, offering a unified approach to managing both containers and virtual machines (VMs). With a supported Red Hat OpenShift operator, the platform is simple to install and manage, becoming a part of the container-based application lifecycle, including cloud-native container management, scheduling, and orchestration.

Red Hat OpenShift Platform Plus

Red Hat OpenShift Platform Plus is an integrated platform that helps you build, modernize, and scale applications. It provides multicluster security, compliance, and comprehensive application and data management, helping ensure consistency across different infrastructures throughout the software supply chain. By offering a complete suite of services, Red Hat OpenShift Platform Plus enables faster development and helps you deliver applications seamlessly across your hybrid cloud environment.

Red Hat OpenShift Platform Plus includes the following products:

- ▶ Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform provides a consistent hybrid cloud foundation that enables organizations to build and scale containerized applications with confidence. With a good track record of supporting business-critical workloads for thousands of customers worldwide, Red Hat OpenShift facilitates seamless cloud migrations and supports the development of innovative customer applications. As a leading contributor to Kubernetes, Red Hat helps ensure that Red Hat OpenShift remains at the forefront of container orchestration.

- ▶ Red Hat Advanced Cluster Management for Kubernetes

Red Hat Advanced Cluster Management for Kubernetes is a tool that helps organizations manage and govern Kubernetes environments at scale. It provides a centralized platform for managing multiple Kubernetes clusters, whether they are deployed on-premises, in public clouds, or at the edge.

For more information, see “Red Hat Advanced Cluster Management for Kubernetes” on page 369.

- ▶ Red Hat Advanced Cluster Security for Kubernetes

Red Hat Advanced Cluster Security for Kubernetes is a security solution for cluster management in the Red Hat environment.

For more information, see “Red Hat Advanced Cluster Security for Kubernetes” on page 369.

- ▶ Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation is a persistent storage and cluster data management solution for Red Hat OpenShift. It offers a distributed, scalable SDS platform that provides enterprise-level data management services.

For more information, see “Red Hat OpenShift Data Foundation” on page 370.

- ▶ Red Hat Quay

Red Hat Quay is a container image registry service that is included as a component of Red Hat OpenShift Platform Plus.

For more information, see “Red Hat Quay” on page 368.

Middleware and application services

This section describes the following services:

- ▶ AMQ Streams
- ▶ Red Hat Fuse
- ▶ Red Hat 3scale API Management
- ▶ GitOps
- ▶ Red Hat Ansible Automation Platform

AMQ Streams

AMQ Streams is a data streaming platform from Red Hat that uses Apache Kafka and Apache ZooKeeper. It is designed for large-scale, high-performance data streaming and processing. AMQ Streams offers scalability by enabling you to add nodes to the cluster to handle increasing loads. It helps ensure high data throughput and low latency, making it suitable for real-time data processing.

The platform provides reliability through features such as data replication and fault tolerance, helping ensure data availability and consistency. It supports various messaging patterns and integrates with applications, endpoints, and devices, enhancing versatility. AMQ Streams also supports multiple programming languages and protocols, making it adaptable for different use cases. Its event-driven architecture is ideal for applications that must respond to data changes in real time, providing a solution for modern data-driven environments.

Red Hat Fuse

Fuse is an open-source integration platform that connects applications, data, and devices across hybrid environments. Built on Apache Camel, it provides a framework for developing, deploying, and managing integration solutions. Fuse supports multiple deployment options, including on-premises, cloud, and hybrid environments, making it versatile for different infrastructure needs. Its cloud-native capabilities enable integration with platforms such as Kubernetes and Red Hat OpenShift, enhancing scalability and flexibility.

One of the key strengths of Fuse is its library of connectors and integration patterns, which simplify connecting disparate systems. Combined with support for microservices architecture, this capability allows organizations to break down monolithic applications into manageable, scalable components. Fuse also supports agile development practices, enabling faster iteration and deployment of integration solutions. By fostering collaboration among integration experts, developers, and business users, Fuse helps ensure that modernization efforts align with business goals and technical requirements.

Red Hat 3scale API Management

3scale API Management is a platform for managing APIs for internal and external users. It enables organizations to share, secure, distribute, control, and monetize APIs on an infrastructure that is built for performance and scalability. The platform uses a hybrid-cloud architecture that separates API management policy execution (traffic managers or API gateways) from policy configuration (API manager). This separation helps ensure that API calls do not route through the 3scale API Management infrastructure, reducing latency and improving performance. 3scale API Management also provides an API administration portal with performance dashboards and developer-facing portals for exposing and documenting APIs. 3scale API Management offers flexible scalability by enabling organizations to add gateways and scale horizontally with demand. It reduces operational costs through automated signup and billing processes and accelerates time to market by simplifying the build-and-deploy process for new features. The platform includes a custom developer portal and interactive API documentation to help developers get started quickly. Its ability to manage APIs efficiently and securely makes it an ideal choice for organizations looking to use APIs for growth and innovation.

GitOps

GitOps is an operational framework that applies DevOps best practices to infrastructure automation by using Git as the single source of truth. It uses Git repositories to manage and control infrastructure through code, enabling automated deployments and seamless rollbacks. By using Git pull requests (PRs), GitOps integrates changes automatically, which helps ensure consistency and reliability in the deployment process. This approach simplifies infrastructure management, reduces manual intervention, and enhances collaboration between development and operations teams.

A key feature of GitOps is its declarative infrastructure, where the wanted system state is defined in Git repositories. This approach helps ensure that all infrastructure changes are tracked and saved as versions, enabling auditing and compliance. GitOps also improves security by using Git's version control system to manage access and changes, reducing the attack surface and enabling quick recovery if there are incidents. The usage of CI/CD pipelines in GitOps workflows further automates deployments, improving efficiency and reducing downtime.

GitOps offers several benefits. It improves productivity by automating repetitive tasks and enabling developers to focus on strategic work. It enhances collaboration by providing a transparent and consistent workflow where all changes go through the same review and approval process. This approach fosters better communication and teamwork. GitOps also reduces operational costs by optimizing resource management and minimizing manual oversight. Overall, GitOps provides an efficient approach to managing cloud-native applications and infrastructure, making it a valuable tool for modern development practices.

Red Hat Ansible Automation Platform

Red Hat Ansible Automation Platform is a solution for streamlining and scaling automation across multiple domains. It provides everything that you need to create, run, and manage automation tasks within a single subscription. The platform includes Ansible Tower, which offers a centralized interface for managing and monitoring automation workflows, and Ansible Galaxy, which provides access to prebuilt automation roles and modules. Ansible Automation Hub offers certified content collections from Red Hat and its partners, helping ensure reliable and secure automation.

One of the key benefits of Ansible Automation Platform is its ability to enhance efficiency and reduce complexity in IT operations. By automating repetitive tasks, it frees time for IT teams to focus on strategic initiatives. The platform supports event-driven automation, enabling real-time responses to changes in the IT environment. It integrates with existing IT infrastructures, providing flexibility and scalability to meet evolving business needs. With features such as automation analytics and Red Hat Insights, organizations gain deeper visibility into automation processes and can optimize performance.

Run times

This section describes several run times that are used in modernization projects.

JBoss

JBoss is an open-source application server that is developed by Red Hat. It is used for building, deploying, and hosting Java-based applications and services. Here are some key features and components of JBoss:

- ▶ Application server: JBoss provides a robust environment for running Java applications, including support for Java Enterprise Edition standards.
- ▶ Modular architecture: It has a modular design, enabling developers to use only the components that they need, which helps optimize performance and resource usage.
- ▶ Integration: JBoss integrates with various development tools and frameworks.
- ▶ Scalability: It supports clustering and load-balancing, which are essential for scaling applications to handle increased traffic and workload.
- ▶ Management and monitoring: JBoss offers tools for managing and monitoring applications, helping ensure that they run smoothly and efficiently.
- ▶ Security: It includes features for securing applications, such as authentication, authorization, and encryption.

JBoss is widely used in enterprise environments for its reliability, flexibility, and strong community support. It plays an important role in modernizing IBM Power servers by enabling migration to more flexible, scalable, and efficient architectures.

Here are some key aspects of how JBoss contributes to this modernization:

- ▶ Integration with Red Hat OpenShift: JBoss, as part of Red Hat's suite, integrates with Red Hat OpenShift on IBM Power servers. This integration enables the deployment of cloud-native applications and the modernization of existing applications without changing the underlying hardware.
- ▶ Microservices architecture: JBoss supports the development of microservices that can be deployed on IBM Power servers. This architecture improves scalability, flexibility, and manageability, making it simpler to adapt to changing customer demands.

- ▶ Containerization: JBoss applications can be containerized by using platforms such as Red Hat OpenShift. Containerization simplifies application management and orchestration, improving resource usage and accelerating deployment cycles.
- ▶ Enhanced performance and security: By using JBoss on IBM Power servers, organizations can benefit from improved performance and enhanced security features. IBM Power servers provide workload isolation and platform integrity, helping ensure that applications run securely and efficiently.
- ▶ Cost savings and efficiency: Running JBoss on IBM Power servers can reduce costs. The combination of JBoss and IBM Power automates and manages infrastructure tasks, enabling teams to focus on higher-value projects.

Overall, JBoss with IBM Power servers provides a framework for modernizing applications, improving performance, and increasing operational efficiency.

JBoss Enterprise Application Platform

JBoss Enterprise Application Platform (JBoss EAP) is a powerful and flexible application server that you can use to build, deploy, and host Java-based applications.

Here are some key features of JBoss EAP:

- ▶ Modular architecture: JBoss EAP uses a modular design that enables you to include only the components that you need. This approach optimizes performance and resource usage.
- ▶ Java EE compatibility: JBoss EAP supports Java EE standards, which makes it suitable for enterprise applications. It includes support for Enterprise JavaBeans (EJB), JavaServer Faces (JSF), and Java Persistence API (JPA).
- ▶ HA and clustering: JBoss EAP provides HA and clustering features to help ensure that applications scale and remain resilient under heavy loads.
- ▶ Management and monitoring: JBoss EAP includes tools for managing and monitoring applications, such as a management CLI and a web-based management console.
- ▶ Security: JBoss EAP offers robust security features, including authentication, authorization, and encryption.
- ▶ Integration: JBoss EAP integrates with various development tools and frameworks to enhance the development and deployment process.

Quarkus

Quarkus is a powerful modernization tool that offers several benefits for enterprises updating applications. As a full-stack, cloud-native Java framework that is developed by Red Hat, Quarkus optimizes Java applications for containerized environments. It supports both Java virtual machine (JVM) and native compilation, which enables faster startup times and lower memory usage. These features make Quarkus ideal for modernizing applications to run efficiently in cloud and Kubernetes environments. By using Quarkus, organizations can transform traditional Java EE applications into lightweight, high-performance microservices, improving scalability and reducing operational costs.

Quarkus integrates seamlessly with IBM Power servers and provides robust support for hybrid cloud deployments. This integration enables enterprises to use IBM Power with high performance and reliability while modernizing their application stack. Quarkus also supports enterprise features such as RESTful APIs, configuration management, service invocation, resilience, security, and monitoring, which makes it a comprehensive solution for application modernization. By adopting Quarkus, businesses can ensure that their applications are future-ready and capable of meeting the demands of modern IT environments.

Red Hat AMQ Broker

Red Hat AMQ is a suite of messaging tools that enables reliable and scalable communication between applications. It includes several components, such as AMQ Broker, AMQ Streams, and AMQ Online, each serving a specific purpose:

- ▶ **AMQ Broker:** Based on Apache ActiveMQ, this component provides robust messaging capabilities and supports multiple protocols, including MQTT, AMQP, and STOMP. It helps ensure reliable message delivery and supports HA and clustering.
- ▶ **AMQ Streams:** Based on Apache Kafka, this component offers a distributed, high-performance data streaming platform that enables real-time data processing and integration.
- ▶ **AMQ Online:** Provides a cloud-native messaging service that enables users to deploy and manage messaging infrastructure in Kubernetes environments.

AMQ plays an important role in modernizing applications. It integrates seamlessly with IBM Power, which enhances the ability to build modern, scalable, and resilient applications. For example, AMQ Streams can run on Red Hat OpenShift Container Platform on IBM Power, enabling high-performance data streaming and real-time processing. This integration enables enterprises to use IBM Power with high performance and reliability while modernizing their messaging infrastructure, helping ensure efficient and secure communication across the IT landscape.

Spring Boot

Spring Boot is an open-source framework that simplifies the development of stand-alone, production-grade Spring-based applications. One of its key features is auto-configuration, which automatically configures the application based on the included dependencies. This feature reduces the need for extensive manual configuration and enables developers to focus on writing business logic instead of boilerplate code. Spring Boot also includes embedded servers such as Tomcat, Jetty, and Undertow, which enable applications to run without deploying WAR files. This embedded server capability streamlines development and deployment, making it easier to start applications quickly.

Another notable feature of Spring Boot is its starter Project Object Models (POMs). These starter POMs simplify dependency management by grouping commonly used dependencies into convenient packages. This approach saves time and helps ensure that the project includes all necessary dependencies for a specific function. Spring Boot also offers production-ready features such as metrics, health checks, and externalized configuration. These features help ensure that the application is ready for production environments and provide insights into application performance and health.

Spring Boot is suitable for developing microservices architectures. Its modular design and ability to integrate with other Spring projects and third-party libraries make it an excellent choice for building scalable and maintainable applications. The framework supports rapid development, which enables developers to create and deploy microservices with minimal configuration. Spring Boot also integrates seamlessly with Spring Cloud, providing tools and frameworks for building robust cloud-native applications.

Common use cases for Spring Boot include building web applications, developing RESTful APIs, and creating microservices architectures. The framework's flexibility and powerful features make it a good choice for modern application development. Whether the goal is to build a simple web application or a complex microservices system, Spring Boot provides the tools and capabilities to achieve these objectives efficiently.

Node.js

Node.js is an open-source, cross-platform runtime environment that runs JavaScript code outside a web browser. It uses the JavaScript 8 engine, the same engine that is used by Google Chrome to run JavaScript code on the server side. Node.js uses an event-driven architecture that efficiently handles asynchronous operations, which makes it ideal for applications that require real-time processing and high concurrency. Its non-blocking I/O model enables it to manage multiple requests simultaneously without waiting for any single operation to complete.

Node.js includes Node Package Manager (NPM), the largest ecosystem of open-source libraries and packages, which simplifies integration and development. This architecture supports the development of scalable applications that are suitable for both small projects and large-scale enterprise applications. Common use cases for Node.js include building web servers and APIs, developing real-time applications such as chat applications and online gaming, and creating microservices architectures. Its performance, unified development process that uses JavaScript for both client-side and server-side, and strong community support make Node.js a good choice for modern web development.

Frameworks such as Express.js and Koa.js enhance Node.js capabilities by providing robust features for web and mobile applications. Node.js is used for its efficiency, scalability, and versatility, which makes it a valuable tool for developers who build high-performance applications.

Node.js provides several advantages when running on IBM Power servers:

- ▶ **Performance:** The non-blocking I/O model in Node.js helps ensure efficient handling of multiple requests, which enhances overall system performance.
- ▶ **Scalability:** The microservices architecture that is supported by Node.js enables applications to scale and meet growing demands.
- ▶ **Cost-effectiveness:** Modernizing with Node.js can be more cost-effective than performing complete rewrites or migrations.

Overall, Node.js provides a robust framework for modernizing IBM Power servers and enables the development of high-performance, scalable, and efficient applications.

Single sign-on

SSO is an authentication scheme that enables users to log in once with a single set of credentials and access multiple applications during the same session.

SSO simplifies user authentication, improves the user experience, and, when implemented correctly, enhances security. It is often used to manage authentication and secure access to company intranets or extranets, student portals, public cloud services, and other environments where users must move between different applications to complete their work. SSO is also increasingly used in customer-facing websites and applications, such as banking and e-commerce platforms, to combine applications from third-party providers into seamless, uninterrupted user experiences.

How single sign-on works

SSO is based on a digital trust relationship between service providers, such as applications, websites, and services, and an identity provider or SSO solution. The SSO solution is often part of a larger [Identity and Access Management \(IAM\)](#) solution.

In general, SSO authentication works as follows:

1. A user logs in to one of the service providers or a central portal (such as a company intranet or college student portal) by using SSO login credentials.
2. When the user is successfully authenticated, the SSO solution generates a session authentication token that contains specific information about the user's identity, such as a username or email address. This token is stored in the user's web browser or in the SSO system.
3. When the user attempts to access another trusted service provider, the application checks with the SSO system to determine whether the user is already authenticated for the session.
 - If the user is authenticated, the SSO solution validates the user by signing the authentication token with a digital certificate, and the user is granted access to the application.
 - If the user is not authenticated, the user is prompted to reenter login credentials.

SSO variations

The SSO process (a single login and set of user credentials providing session access to multiple related applications) that is described in “How single sign-on works” on page 376 is sometimes called simple SSO or pure SSO. Other types of SSO include the following ones:

- ▶ Adaptive SSO: Requires an initial set of login credentials but prompts for other authentication factors or a new login when more risks emerge, such as when a user logs in from a new device or attempts to access sensitive data or functions.
- ▶ Federated identity management (FIM): A superset of SSO. SSO is based on a digital trust relationship among applications within a single organization's domain, and FIM extends that relationship to trusted third parties, vendors, and other service providers outside the organization. For example, FIM might enable a logged-in employee to access third-party web applications (such as Slack or Webex) without an extra login or with a simple username-only login.
- ▶ Social login: Enables users to authenticate with applications by using the same credentials that they use for social media sites. For third-party application providers, social login can discourage unwanted behaviors, such as false logins or shopping cart abandonment, and provide valuable information for improving their applications.

Related technologies

SSO can be implemented by using several authentication protocols and services:

- ▶ Security Assertion Markup Language (SAML) and SAML 2.0: SAML is the longest-standing open standard protocol for exchanging encrypted authentication and authorization data between an identity provider and multiple service providers. Because it provides greater control over security than other protocols, SAML is typically used to implement SSO within and between enterprise or government application domains.
- ▶ OAuth and OAuth 2.0: OAuth is an open standard protocol that exchanges authorization data between applications without exposing the user's password. OAuth enables a single login to streamline interactions between applications that would typically require separate logins. For example, OAuth makes it possible for LinkedIn to search email contacts for potential new network members.

- ▶ OpenID Connect (OIDC): Another open standard protocol, OIDC uses [REST APIs](#) and JSON authentication tokens to enable a website or application to grant users access by authenticating them through another service provider. Layered on top of OAuth, OIDC is used primarily to implement social logins for third-party applications, shopping carts, and more. A lighter-weight implementation, OIDC is often preferred over SAML for implementing SSO across [SaaS](#) and cloud applications, mobile apps, and [Internet of Things \(IoT\)](#) devices.
- ▶ Lightweight directory access protocol (LDAP): LDAP defines a directory for storing and updating user credentials and a process for authenticating users against the directory. Introduced in 1993, LDAP remains the authentication directory solution of choice for many organizations implementing SSO because it provides granular control over directory access.
- ▶ Active Directory Federation Services (ADFS): ADFS runs on Microsoft Windows Server to enable FIM, including SSO, with on-premises and off-premises applications and services. ADFS uses Active Directory Domain Services (ADDS) as an identity provider.

Benefits of SSO

SSO saves users time and effort. For example, instead of logging in to multiple applications several times per day, corporate users can log in to the corporate intranet once for all-day access to every application they need. By reducing the number of passwords that users must remember and the number of user accounts that administrators must manage, SSO improves both user experience and administrative efficiency.

SSO can provide many other benefits.

- ▶ Reduced password fatigue

Users who manage multiple passwords often adopt risky habits, such as reusing short, weak passwords or creating slight variations. If a hacker compromises one password, they can access multiple applications. SSO enables users to replace multiple weak passwords with one strong password that is easier to remember and harder for hackers to break.

- ▶ Fewer password- and credential-related vulnerabilities

According to the [IBM X-Force® Threat Intelligence Index 2024](#), cyberattacks that used stolen or compromised credentials increased by 71% in 2023 compared to the previous year. SSO reduces or eliminates the need for password managers, spreadsheets, sticky notes, and other memory aids, all of which create opportunities for hackers or accidental exposure.

- ▶ Fewer help desk calls

In an HDI survey, up to 30% of help desk calls are related to password resets¹. Most SSO solutions enable users to reset passwords without help desk assistance.

- ▶ Simplified security management

SSO provides administrators with centralized control over account provisioning and access permissions. When a user leaves the organization, administrators can revoke permissions and decommission accounts in fewer steps.

- ▶ Improved regulatory compliance

SSO helps organizations meet regulatory requirements for protecting [personal identity information \(PII\)](#) and controlling data access. It also supports compliance with specific regulations, such as HIPAA, which mandate session timeouts.

¹ Source: <https://www.thinkdhi.com/library/supportworld/2011/password-reset-practices>

SSO security risks

The main risk of SSO is that if an attacker compromises a user's credentials, they can gain access to most or all applications and resources on the network. Requiring users to create long, complex passwords and encrypting and protecting those passwords wherever they are stored reduces the likelihood of this scenario.

Most security experts recommend implementing [two-factor authentication \(2FA\)](#) or multi-factor authentication (MFA) as part of any SSO deployment. 2FA or MFA requires users to provide at least one more authentication factor beyond a password, such as a passcode that is sent to a mobile device, a fingerprint, or an ID card. Because these additional factors are difficult for attackers to steal or spoof, MFA reduces the risks that are associated with compromised SSO credentials.

IBM

This section highlights IBM solutions to help you modernize your infrastructure and applications.

System management

Modernizing your infrastructure often requires new approaches to system management. These approaches include using AI-enhanced observability, Application Resource Management (ARM), and cloud-ready solutions to help you refactor applications.

IBM Power Private Cloud with Shared Utility Capacity

IBM Power Private Cloud with Shared Utility Capacity (also known as Power Enterprise Pools 2.0 (PEP2)) provides enterprises with cloud-like flexibility and efficiency while maintaining the security and control of on-premises infrastructure. This solution enables organizations to share resources across multiple IBM Power servers and optimizes usage through a pay-per-use model with minute-level metering. Processor cores and memory are dynamically allocated as needed, reducing upfront costs and improving operational agility.

One of the key features of PEP2 is its integration with the IBM Cloud Management Console (CMC). CMC provides a centralized interface for managing resource pools, monitoring usage, and handling capacity credits. This integration enables enterprises to manage IT infrastructure efficiently and respond quickly to changing business needs.

Integration with Red Hat OpenShift enables seamless deployment and management of containerized applications on IBM Power servers. This integration enhances scalability and accelerates modernization efforts, making the platform ideal for organizations adopting hybrid cloud strategies and modernizing IT environments without sacrificing performance or control.

Turbonomic

Turbonomic is a leading solution for ARM, which is a hierarchical, application-driven approach that continuously analyzes application resource requirements and generates fully automatable actions to help ensure optimal performance. Turbonomic runs continuously and scales to support the largest and most complex environments.

To perform ARM, Turbonomic represents the environment as a supply chain of resource buyers and sellers that work together to meet application demand. Buyers, such as VMs, receive a budget to acquire the resources that applications need to perform. Sellers, such as hosts, price their available CPU, memory, storage, and other resources based on real-time usage. This approach keeps applications in an optimal state.

Turbonomic is a microservices-based platform that runs either in your network or in the public cloud. It discovers and monitors the application environment through defined targets.

Turbonomic then analyzes the environment, anticipates performance or efficiency risks, and recommends actions to prevent issues before they occur.

For more information about Turbonomic, see [IBM Turbonomic Application Resource Management documentation](#).

Instana

IBM Instana Observability is a platform that helps you analyze and troubleshoot microservices and containerized applications. It provides automated application performance monitoring, end-user experience monitoring (EUM), root cause analysis, and anomaly detection. Instana gives you visibility into the health and performance of your applications and services.

Instana automatically discovers applications and services, provides context for the observed data, and enables you to take intelligent action based on that information.

- ▶ Automates discovery and visibility

Instana automatically discovers and monitors applications, services, infrastructure, web browsers, mobile applications, and more for over 200 domain-specific technologies. It displays real-time data through distributed tracing and 1-second metrics.

- ▶ Provides context

Instana automates dependency mapping across the full stack to deliver flexible application perspectives and powerful analytics. It places performance data in context to enable rapid issue prevention and remediation. You can drill down into the full repository of application request trace data to generate insights with unlimited flexibility.

- ▶ Enables intelligent decision-making

Instana alerts you within seconds when performance or stability issues impact customers. It automates root cause analysis by correlating events, performance thresholds, errors, changes, and service-level agreement (SLA) violations.

For more information about Instana, see [IBM Instana Observability](#).

IBM Cloud Pak

IBM Cloud Pak is a set of integrated software solutions that help businesses accelerate their digital transformation. Built on Red Hat OpenShift, IBM Cloud Pak provides a robust, cloud-native platform that enables enterprises to modernize applications, manage workloads, and optimize resources across hybrid cloud environments. Each IBM Cloud Pak is a comprehensive, pre-integrated package of software, tools, and services that addresses specific business needs, such as data management, AI, security, and automation.

IBM Cloud Pak is available in several offerings, each tailored to specific use cases. This section describes the following IBM Cloud Pak offerings:

- ▶ IBM Cloud Pak for Automation (CP4A)
- ▶ IBM Cloud Pak for Integration (CP4I)

- ▶ IBM Cloud Pak for Business Automation (CP4BA)
- ▶ IBM Cloud Pak for AIOps
- ▶ IBM Cloud Pak for Data (CP4D)

IBM Cloud Pak offerings are built with open-source technologies and designed for hybrid cloud environments. They operate across on-premises infrastructure and public clouds. These offerings provide key benefits, such as simplifying IT management, enhancing scalability, reducing operational costs, and improving security and compliance. By using Red Hat OpenShift as the underlying platform, IBM Cloud Pak offerings help ensure a consistent and flexible environment for deploying and managing applications, regardless of where they are hosted.

CP4A

CP4A is an integrated suite of tools that helps businesses automate processes across the organization, including business operations, IT workflows, customer interactions, and data management. CP4A combines several automation capabilities, such as Robotic Process Automation (RPA), business process management (BPM), workflow automation, decision automation, and content management, into one comprehensive platform. By streamlining and automating these processes, businesses can increase efficiency, reduce operational costs, improve accuracy, and enhance customer experiences.

Here are some use cases for CP4A:

- ▶ Customer service automation
- ▶ Invoice and document processing
- ▶ Human resources automation
- ▶ Supply chain management
- ▶ Fraud detection and prevention

CP4A is a comprehensive solution that helps businesses automate a wide range of processes to enhance operational efficiency, agility, and decision-making. With tools for RPA, BPM, decision automation, content management, and AI-powered insights, CP4A enables organizations to optimize workflows, reduce costs, improve compliance, and deliver better customer experiences. Its cloud-native architecture and scalability enable businesses to meet the growing demands of the digital age while maintaining flexibility across hybrid and multi-cloud environments.

CP4I

CP4I is an integrated suite of tools and services that helps businesses streamline and simplify integration processes across hybrid and multi-cloud environments. By providing a unified, flexible, and scalable platform, CP4I enables organizations to connect applications, data, and services across different systems, whether on-premises, in the cloud, or in hybrid environments. This approach helps ensure seamless data flow, improves business agility, and accelerates the development of new digital capabilities.

Here are some use cases for CP4I:

- ▶ Hybrid cloud integration
- ▶ Customer experience transformation
- ▶ Supply chain optimization
- ▶ IoT integration
- ▶ Banking and financial services integration

CP4I is a comprehensive, flexible, and scalable platform that helps businesses integrate data, applications, and services across hybrid and multi-cloud environments. By combining API management, application integration, enterprise messaging, event streaming, and data integration into a single platform, CP4I enables organizations to accelerate digital transformation, improve operational agility, and deliver better customer experiences. With built-in security features and scalability, CP4I provides a solution that evolves with the needs of modern businesses, helping them remain competitive in an increasingly digital and interconnected world.

CP4BA

CP4BA is an integrated suite of AI-powered tools and solutions that helps businesses streamline and automate operations, improve efficiency, and drive innovation. It combines capabilities such as BPM, workflow automation, (RPA, decision automation, and content management within a unified platform. CP4BA enables organizations to automate repetitive tasks, make data-driven decisions, and adapt to changing market conditions by simplifying complex business processes.

Here are some use cases for CP4BA:

- ▶ Customer service automation
- ▶ Supply chain optimization
- ▶ Claims processing
- ▶ Human resources automation
- ▶ Compliance and risk management

CP4BA is a powerful platform that enables businesses to drive digital transformation by automating business processes. By combining AI, RPA, BPM, and decision automation in a unified platform, CP4BA helps organizations become more efficient, agile, and data-driven. It enables businesses to reduce costs, enhance productivity, and improve decision-making while scaling automation across hybrid cloud environments. Whether improving customer service, optimizing supply chains, or helping ensure compliance, CP4BA provides the flexibility and intelligence that are needed to adapt to evolving business demands.

IBM Cloud Pak for AIOps

IBM Cloud Pak for AIOps is an integrated, AI-powered platform that helps businesses automate IT operations (ITOps) and improve system performance, availability, and security. By using AI and machine learning (ML) technologies, IBM Cloud Pak for AIOps provides proactive insights, intelligent automation, and predictive analytics. These capabilities enable IT teams to detect and resolve issues faster, optimize performance, and prevent downtime.

Here are some use cases for IBM Cloud Pak for AIOps:

- ▶ Application Performance Management
- ▶ Infrastructure and network optimization
- ▶ Security operations
- ▶ Hybrid cloud management

IBM Cloud Pak for AIOps is a transformative solution that brings AI and automation to IT operations (ITOps), enabling organizations to proactively monitor, manage, and optimize their IT infrastructure. By using ML and predictive analytics, the platform helps businesses detect and resolve issues faster, improve system performance, and reduce downtime. Whether for Application Performance Management, network optimization, or security operations, IBM Cloud Pak for AIOps enables organizations to create a smarter, more resilient IT environment that drives better business outcomes.

CP4D

CP4D is an integrated data and AI platform that helps organizations accelerate data-driven decision-making by unifying data management, analytics, AI, and automation in a single solution. Designed for hybrid and multi-cloud environments, it enables businesses to collect, organize, and analyze data at scale while helping ensure consistency, governance, and security across all data assets. CP4D simplifies the management of data pipelines, ML models, and analytics, making it easier for teams to derive insights and deliver business value.

Here are some use cases for CP4D:

- ▶ Data modernization
- ▶ AI and predictive analytics
- ▶ Data-driven decision making
- ▶ Compliance and risk management

CP4D provides a comprehensive, integrated platform that helps organizations manage and use data more effectively. It combines data management, AI, and analytics tools with robust governance and security features to enable data-driven decision-making across the enterprise. By simplifying the complexity of working with data at scale, CP4D accelerates AI adoption, improves business agility, and enhances operational efficiency in hybrid cloud environments. Whether modernizing systems, developing AI models, or optimizing business processes, CP4D provides the tools that are needed to unlock the full potential of data.

IBM watsonx

IBM watsonx is IBM's portfolio of AI products that accelerates the impact of generative AI in core workflows to improve productivity.

- ▶ Open your AI future.

Get the flexibility that you need to make the right AI choices for your business. Choose an open-source foundation model, bring your own, or use existing models, and run it across any cloud.

- ▶ Trust your AI outputs.

Create responsible AI with trusted enterprise data and governed processes. Use open, transparent technology. Employ governance and security controls for compliance.

- ▶ Integrate to innovate.

Deploy AI with minimal disruptions to your systems or operations. Embed it into specific use cases to realize value quickly. Transform processes to increase productivity.

IBM watsonx solutions

IBM watsonx provides a comprehensive portfolio of AI products to improve productivity across core workflows:

- ▶ IBM watsonx Orchestrate®

Create, deploy, and manage AI assistants and agents to automate and simplify business and customer-facing processes.

- ▶ IBM watsonx Code Assistant

Accelerate developer productivity and reduce time to market by infusing AI into the application lifecycle to automate development tasks and streamline workflows.

- ▶ IBM watsonx.ai
Develop custom AI applications faster with an integrated, collaborative developer studio that includes an AI toolkit and full AI lifecycle management.
- ▶ IBM watsonx.data®
Manage, prepare, and integrate trusted data from any source and format to unlock AI insights faster and improve the relevance and precision of AI applications.
- ▶ IBM watsonx.governance®
Automate governance to proactively manage AI risks, simplify regulatory compliance, and create responsible, explainable AI workflows.

Open-source solutions

Open-source solutions on IBM Power servers provide a powerful, flexible foundation for modern workloads, enabling the use of technologies such as Linux, Kubernetes, and container orchestration platforms like Red Hat OpenShift. IBM Power supports several open-source databases, analytics tools, and development frameworks that are optimized for the architecture's high performance and scalability. By combining the reliability and security of IBM Power with the innovation and cost-efficiency of open-source software (OSS), organizations can build agile, cloud-ready environments that support digital transformation, DevOps practices, and hybrid cloud strategies.

Kernel-based Virtual Machine

Kernel-based Virtual Machine (KVM) is an open-source virtualization technology that enables Linux-based operating systems (OSs) to run VMs on hardware. KVM is part of the Linux kernel and turns it into a hypervisor, enabling the OS to run multiple isolated virtual environments on a single physical machine. KVM enables organizations to virtualize IT infrastructure and run different guest operating systems, including Linux, Windows, and others, alongside the host OS.

On IBM Power10, KVM is introduced as a virtualization option within a PowerVM logical partition (LPAR). You first create a Linux LPAR using PowerVM, and then run KVM within that LPAR to host other Linux VMs. KVM on IBM Power10 combines the advantages of open-source virtualization with the high performance, scalability, and enterprise-grade features of IBM Power10 architecture. This integration provides a powerful solution for businesses that require virtualization at scale, offering enhanced performance, flexibility, and security. Running KVM on IBM Power10 enables organizations to use advanced hardware while benefiting from open-source, high-performance virtualization capabilities.

Artificial intelligence solutions

The ability to use AI to gain customer insights or manage complex environments is critical to modernization efforts. This section describes AI software solutions that use the AI acceleration technologies that are built into IBM Power hardware.

RocketCE and Rocket AI Hub for IBM Power

RocketCE for IBM Power is a set of AI and data science packages that are built and supported for the IBM Power Architecture. The packages include open-source AI tools that are optimized for IBM Power10, which uses on-chip acceleration. To download the packages, see [Rocket Software's public Anaconda channel](#).

Rocket AI Hub for IBM Power is an integrated and available set of open-source AI platform tools that are optimized for IBM Power, which includes the following tools:

- ▶ Katib
- ▶ Kubeflow
- ▶ Kubeflow Pipelines
- ▶ KServe
- ▶ RocketCE

All tools are delivered as container images that run in Kubernetes-based environments such as Red Hat OpenShift. They are integrated through Kubeflow, optimized to use the AI hardware acceleration capabilities of the IBM Power platform, and include optional commercial support.

Ollama

Ollama is an open-source project that enables you to run large language models (LLMs) on your own hardware. It supports a range of models that are available in the Ollama library, including the IBM Granite dense family. The IBM Granite 2B and 8B models are text-only dense LLMs trained on more than 12 trillion tokens. They are designed to support tool-based use cases and retrieval-augmented generation (RAG), streamlining tasks such as code generation, conversion, and bug fixing.

Table A-1 describes some of the models that are supported by Ollama.

Table A-1 Models supported by Ollama

Model	Parameters	Download and run
granite3.1-dense	8B	ollama run granite3.1-dense:8b
granite3.1-dense	2B	ollama run granite3.1-dense:2b
Llama 3.3	70B	ollama run llama3.3
Llama 3.2	3B	ollama run llama3.2
DeepSeek-R1	7B	ollama run deepseek-r1
DeepSeek-R1	671B	ollama run deepseek-r1:671b

To use Ollama, download the correct binary file from [Ollama](#) and can install it on your hardware.

To start the Ollama service, run the command that is shown in Example A-1.

Example: A-1 Starting the Ollama service

```
# ollama serve
```

To pull and run a model (you can find the models at [Ollama](#)), run the command that is shown in Example A-2.

Example: A-2 Running a model

```
# ollama run granite3.1-dense.8b
>>>
```

Ollama creates an isolated environment to run LLMs locally and includes all the components that are required for deploying AI models, such as model weights, configuration files, and dependencies.

Here are some examples of how Ollama can help:

- ▶ Creating local chatbots

Developers can use Ollama to create highly responsive AI-driven chatbots that run entirely on local servers, helping ensure that customer interactions remain private.

- ▶ Conducting local research

Universities and data scientists can use Ollama to conduct offline machine learning research. This approach enables experimentation with datasets in privacy-sensitive environments, helping ensure that the work remains secure and not exposed to external parties.

- ▶ Building privacy-focused AI applications

Ollama provides an ideal solution for developing AI applications for businesses that handle sensitive information.

- ▶ Integrating AI into existing platforms

Ollama integrates easily with existing software platforms, enabling businesses to add AI capabilities without overhauling current systems.

Streamlit

Streamlit is an open-source Python framework that enables data scientists and AI and ML engineers to build dynamic data applications with minimal code. It offers features for creating data science applications, including support for data visualization, ML, and deep learning. Streamlit also provides tools for deploying and sharing applications, making it simple to collaborate and share projects.

Streamlit is flexible and scalable, supporting a wide range of applications from simple data analysis tools to complex ML models. Its flexibility makes it suitable for both experienced and novice developers.

Common use cases include building data science projects, ML models, and web applications. These capabilities make Streamlit a choice for developing interactive, data-driven applications.

Llama.cpp

Llama.cpp is a C/C++ library that is designed for efficient inference of LLMs. It can use the high performance and scalability of IBM Power servers. Llama.cpp supports processing GGML-formatted models, such as Meta's LLaMA, Vicuna, and Wizard, without requiring a GPU.

To set up llama.cpp on IBM Power10, ensure that prerequisites such as gcc-toolset-13 are installed. The library can be built from source by using CMake, and it supports various BLAS back ends, including OpenBLAS, for optimized performance. Once configured, llama.cpp can efficiently tokenize prompts and generate responses by using top-K and top-P sampling algorithms. This setup enables enterprises that use IBM Power for AI workloads, improving efficiency and scalability for AI applications².

² Source:

<https://community.ibm.com/community/user/blogs/amrita-h-s/2025/01/30/how-to-run-a-l1m-model-on-ibm-power10-using-llamac>

Trovares

Rocketgraph (formerly Trovares) offers the xGT graph analytics platform, which is highly compatible with IBM Power servers. Rocketgraph xGT is designed to handle large and complex graph problems, enabling enterprises to build a single property graph from existing data stores and scale to datasets with hundreds of billions of edges.

Running Rocketgraph xGT on IBM Power10 servers provides performance advantages. IBM Power10 servers, with their superscalar, multithreaded, and multi-core architecture and embedded AI acceleration technology, are ideal for high-performance workloads. Performance tests show that Rocketgraph xGT on IBM Power E1050 servers deliver faster results than x86 servers, making it an excellent choice for enhancing cybersecurity operations, fraud detection, and supply chain optimization³.

Equitus AI

Equitus AI is a US-based technology company that specializes in AI-powered graph data analysis for government and commercial clients. Its platforms include KGNN for unifying data into intelligent knowledge graphs and EVS for real-time video analytics. The company focuses on delivering accurate, traceable, and secure AI solutions for complex data challenges.

KGNN runs natively on IBM Power10 servers, enabling organizations to create autonomous AI systems that operate at the edge, independent of external cloud services, and without GPU resource constraints. The on-premises Equitus KGNN appliance provides a robust solution for unifying and connecting knowledge assets across diverse systems and applications. This integration enables a holistic understanding of complex relationships and supports enhanced decision-making.

OpenTech

OpenXAI-Opentech is a Saudi Arabian technology company that specializes in on-premises AI chatbot platforms (OpenXAI) and custom drones. The company prioritizes data security and uses IBM Power for its AI infrastructure. It also develops AI solutions for smart home applications. OpenTech uses IBM Power infrastructure, particularly Power10 servers, to deliver AI-powered chatbot solutions with a focus on performance, security, and the ability to operate on-premises for data-sensitive applications.

ElinarAI

ElinarAI is an advanced AI solution that automates manual processes and enhances data management across various industries. It uses cognitive AI to interpret and transfer information, reducing errors and improving efficiency. ElinarAI can perform tasks that require human cognition, such as processing large volumes of data, handling sensitive material, and conducting complex analyses.

ElinarAI provides specialized capabilities to automate the analytics of unstructured data. It enables the creation of Investigation-Specific AIs (ISAs) to scan and recognize entities on a large scale with high accuracy. This capability is useful for law enforcement and intelligence communities that manage extensive datasets, such as in cases of money laundering, tax evasion, and fraud.

ElinarAI has a strong relationship with IBM. Its solutions often augment IBM products such as CP4A and IBM Watson Discovery. The platform is also optimized to run on the IBM Power infrastructure.

³ Source:

<https://community.ibm.com/community/user/blogs/jenna-murillo/2025/02/18/introducing-rocketgraph-xgt-on-ibm-power>

Wallaroo

Wallaroo AI is a platform that simplifies and accelerates the deployment and management of ML models in production. It helps businesses use AI by providing tools for the following functions:

- ▶ Deployment: Streamlines the process of moving models into production environments. Wallaroo supports various model frameworks, hardware architectures, and cloud environments. It automates model packaging and offers self-service deployment capabilities.
- ▶ Inference: Runs models efficiently to generate predictions. Wallaroo focuses on high-performance batch and real-time inference with optimized resource usage.
- ▶ Observability: Monitors model performance and identifies issues. Wallaroo provides tools for tracking metrics, detecting drift, and generating reports and alerts.
- ▶ Scalability: Handles increasing volumes of data and traffic. Wallaroo is designed to scale AI deployments to meet real-world application demands.

Wallaroo addresses the challenges of using AI, enabling organizations to use ML effectively for practical use cases. It emphasizes efficiency and flexibility, with features such as automated workflows, support for diverse use cases, and integrated tools.

For more information, see [Wallaroo](#).

Finacle AI

IBM and Finacle have maintained a strong relationship for more than 20 years. IBM Power serves as a trusted infrastructure for Finacle, with 70% of global Finacle deployments running on IBM Power servers.

As banks work to achieve strategic objectives, such as improving customer experience, drive personalization, enhancing operational efficiency, and accelerate speed to market, they increasingly turn to AI to meet these goals effectively.

To help customers adopt AI, Finacle introduced the Finacle AI Platform, which is an innovative solution that is designed with explainability and security. The platform attaches to the core banking application, enabling customers to seamlessly integrate AI into digital operations and derive actionable insights.

The Finacle AI Platform accelerates AI adoption by providing a banking-specific AI framework that offers the following capabilities:

- ▶ Low-code solution: Enables business users to use AI without extensive technical expertise.
- ▶ Pre-built predictive analytics use cases: Supports different business lines, user roles, and customer scenarios.
- ▶ Custom AI services: Provides design, build, and deployment services for tailored AI use cases.

Deploying the Finacle AI Platform on IBM Power simplifies integration with core banking systems, lifecycle management, capacity management, security, and compliance.

Figure A-2 on page 389 shows a deployment architecture on IBM Power.

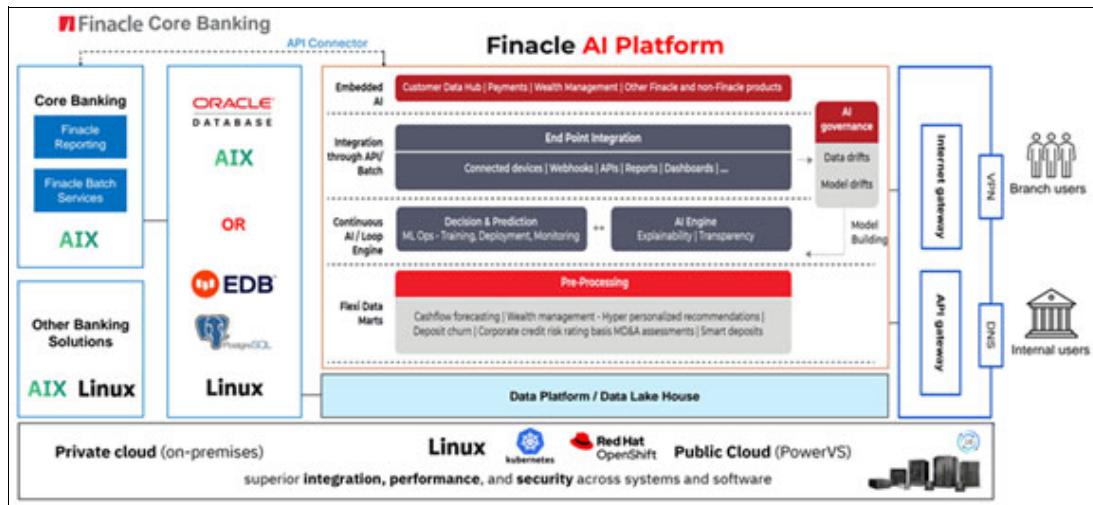


Figure A-2 Sample deployment on the IBM Power for Finacle AI platform

Pipelines

Pipelines are automated workflows that are used in continuous integration and continuous delivery (CI/CD) to streamline and standardize the process of building, testing, and deploying applications. They automatically integrate code changes, run tests, and deliver updates to production environments, reducing manual effort and minimizing the risk of errors.

A typical pipeline includes stages such as source code integration, automated testing, artifact generation, and deployment. These stages help teams deliver software faster and more reliably. By implementing CI/CD pipelines, organizations can improve efficiency, consistency, and agility throughout the development lifecycle.

This section explores CI/CD pipelines that are commonly used to modernize an IBM Power infrastructure.

GitHub

GitHub is a web-based platform that provides version control through Git, enabling developers to collaborate on projects effectively. It offers features such as repositories, branches, PRs, and issues to manage and track code changes. GitHub's collaborative tools enable multiple developers to work on the same project, review code, and merge changes seamlessly.

GitHub integrates with various CI/CD tools, making it simpler to automate workflows and deploy applications. Beyond version control, GitHub fosters a global developer community that contributes to open-source projects, shares knowledge, and collaborates on innovative solutions.

The platform supports multiple programming languages and frameworks, making it versatile for different projects. GitHub also provides GitHub Actions, an automation tool that enables developers to create custom workflows directly within repositories. With extensive documentation and a simple interface, GitHub has become an essential tool for modern software development.

GitLab

GitLab is an All-in-One (AIO) DevOps platform that streamlines the software development lifecycle by integrating tools for source code management, CI/CD, security, and monitoring into a single application. It enhances team collaboration through features such as code review, issue tracking, and project management. Its CI/CD capabilities automate testing and deployment to help ensure rapid, reliable code delivery.

GitLab also includes advanced security features such as vulnerability scanning and compliance checks to safeguard applications throughout their lifecycle. As an open-source platform, GitLab benefits from a global developer community that contributes to its ongoing improvement.

Its flexibility enables users to tailor the platform to unique workflows and integrate with a wide range of external tools and services. With an intuitive user interface (UI) and extensive documentation, GitLab is a trusted choice for organizations that aim to accelerate DevOps adoption, boost productivity, and foster seamless collaboration across development teams.

GitHub and GitLab share many characteristics because both are based on Git and provide similar features. However, there are several key differences:

- ▶ GitLab is open source, which enables you to download the source code and self-host the service on your own servers or a cloud provider.
- ▶ GitLab offers its own deployment platform, which is built on Kubernetes. With GitHub, you must use an external platform, such as Amazon Web Services (AWS) or Heroku, and trigger deployments there.
- ▶ GitLab includes built-in CI/CD and DevOps workflows, making it a comprehensive solution for the entire software development lifecycle. GitHub requires integration with third-party CI/CD tools such as Jenkins, CircleCI, or Travis CI.

Bitbucket

Bitbucket is a Git-based source code repository hosting service that is owned by Atlassian. It provides a central platform for managing Git repositories, collaborating on code, and guiding development workflows. Bitbucket integrates seamlessly with other Atlassian tools such as Jira and Trello, enhancing project management and team collaboration.

The platform offers features such as PRs, code reviews, and built-in CI/CD through Bitbucket Pipelines, which enables teams to automate build, test, and deployment processes. Bitbucket supports both cloud and self-managed data center hosting options, catering to different organizational needs.

With its simple UI and strong collaboration tools, Bitbucket is a suitable choice for teams that use Atlassian's DevOps ecosystem.

Databases

Modern databases are essential for digital transformation. They enable businesses to store, manage, and analyze large volumes of data efficiently. These databases power applications, support real-time decision-making, and drive innovation through features such as scalability, flexibility, and support for diverse data types.

By providing the foundation for data-driven insights, modern databases help organizations optimize operations, enhance customer experiences, and gain a competitive advantage in a data-driven world.

MongoDB

MongoDB is an open-source, non-relational database management system (DBMS) that uses flexible documents instead of tables and rows to process and store data. As a NoSQL database solution, MongoDB does not require a relational schema. It provides an elastic data storage model that enables users to store and query multiple data types with ease. This approach simplifies database management for developers and creates a highly scalable environment for cross-platform applications and services.

MongoDB uses documents or collections of documents as the basic units of data. These documents are formatted as Binary JSON (BSON) and can store various data types while being distributed across multiple systems. Because MongoDB employs a dynamic schema design, users have flexibility when creating data records, querying document collections through MongoDB aggregation, and analyzing large volumes of information.

Over the years, MongoDB has become a trusted solution for businesses that need a powerful and highly scalable NoSQL database. MongoDB offers more than traditional document-based storage and includes several capabilities that make it stand out:

- ▶ Load-balancing: MongoDB uses a sharding process to distribute large datasets across multiple VMs while maintaining acceptable read/write throughput. This horizontal scaling approach helps organizations avoid the cost of vertical hardware scaling while expanding the capacity of cloud-based deployments.
- ▶ Ad hoc queries: MongoDB supports ad hoc queries without requiring a predefined schema, providing flexibility for dynamic data requirements.
- ▶ Multi-language support: MongoDB supports several programming languages, including Python, PHP, Ruby, Node.js, C++, Scala, and JavaScript.

MongoDB editions

MongoDB is available in two main editions: Community Edition and Enterprise Edition.

- ▶ MongoDB Community Edition: A no-charge, open-source version that is suitable for learning and development. It includes core database functions.
- ▶ MongoDB Enterprise Edition: A commercial version that builds on the Community Edition by adding advanced features for mission-critical deployments:
 - Enhanced security: LDAP and Kerberos integration, auditing, and encryption at rest (with KMIP) to meet stringent security requirements.
 - Advanced management tools: MongoDB Ops Manager automates deployment, upgrades, backups, and monitoring.
 - More features: Includes the BI Connector for integration with SQL-based business intelligence tools.
 - Support: Provides 24x7 technical support.

MariaDB

MariaDB is a community-developed, open-source RDBMS that is compatible with MySQL. It is fast, reliable, and simple to use. Built on the Linux kernel, MariaDB provides features such as multi-user support, concurrent access, advanced indexing, and transactional capabilities.

Here are the key features of MariaDB:

- ▶ High performance: Known for speed and efficiency, MariaDB is suitable for demanding workloads.
- ▶ Robustness: Offers a stable and reliable platform with a good track record.
- ▶ Active community: Supported by a large and active community that provides resources, support, and ongoing development.

MariaDB is suitable for several applications, including web applications, e-commerce platforms, and data analytics solutions. Its speed and performance make it ideal for handling large data volumes, and its flexibility appeals to both experienced and novice database administrators.

PostgreSQL

PostgreSQL is a powerful, open-source object-relational database system that extends the SQL language with advanced features to store and scale complex data workloads. It provides capabilities that help developers build applications, enable administrators to protect data integrity, and support fault-tolerant environments for managing datasets of any size.

Here are some key use cases for PostgreSQL:

- ▶ Online transaction processing (OLTP) and analytics: PostgreSQL is a general-purpose OLTP database that is suitable for use cases such as e-commerce, Customer Relationship Management (CRM), and financial ledgers. Its SQL compliance and query optimizer also make it effective for general-purpose analytics.
- ▶ Geographic information systems: PostgreSQL supports geospatial data through the PostGIS extension, which complies with Open Geospatial Consortium (OGC) standards. This capability enhances situational awareness, improves fraud detection, and supports advanced data analysis.
- ▶ Database consolidation: Organizations can migrate databases to PostgreSQL to reduce licensing costs, retire servers, and eliminate database sprawl. This approach decreases total cost of ownership (TCO), removes vendor lock-in, and improves application portability.

PostgreSQL is an ideal database solution for enterprises across various industries due to its flexibility, scalability, and robust feature set.

Fujitsu Enterprise Postgres

Fujitsu Enterprise Postgres is an enhanced, enterprise-grade PostgreSQL solution that is designed for organizations that require strong query performance and HA. It builds on the open-source PostgreSQL system by adding features for improved security, performance, and management.

Here are some key enhancements of Fujitsu Enterprise Postgres:

- ▶ Enhanced security: Includes tools such as Transparent Data Encryption and Data Masking.
- ▶ HA: Provides preconfigured clusters with automated failover.
- ▶ Performance optimizations: Features such as Vertical Clustered Index and Global Meta Cache improve query performance.

Fujitsu Enterprise Postgres delivers a robust, cost-effective database solution with enterprise-level support.

Neo4j

Neo4j is a leading graph database management system (DBMS) that is designed to efficiently handle and query large, complex datasets in the form of graphs. Unlike traditional relational databases that organize data in tables and rows, Neo4j stores data as nodes (representing entities) and relationships (representing connections between entities). This structure enables intuitive and flexible data modeling, making Neo4j ideal for applications where relationships are central, such as social networks, fraud detection, and recommendation systems.

Neo4j uses Cypher, a query language that is optimized for graph operations, enabling users to express complex queries that traverse relationships easily and intuitively.

One of the key advantages of Neo4j is its ability to handle highly connected data with superior performance, even as the dataset scales. Graph databases like Neo4j can efficiently run queries that involve multiple hops or complex relationships, which are slow and cumbersome in traditional relational databases. Neo4j is suitable for use cases such as network analysis, real-time recommendation engines, and knowledge graphs.

Furthermore, Neo4j's ACID-compliant transactional nature helps ensure data consistency and reliability, which is critical in applications requiring real-time insights from large, interconnected datasets. Its scalability and rich set of features make it a powerful tool for modern applications using graph-based data.

EnterpriseDB

EnterpriseDB (EDB) is a leading provider of enterprise-class PostgreSQL solutions. It offers robust database management tools and services. The flagship product, EDB Postgres Advanced Server, enhances PostgreSQL with features such as performance optimization, security improvements, and Oracle database compatibility. EDB also provides tools for high availability and disaster recovery (HADR) and monitoring. These tools help organizations maintain reliable and efficient database operations.

EDB solutions support modern application development and deployment, including integration with cloud-native technologies such as Kubernetes and Red Hat OpenShift. These integrations enable organizations to use containerization and orchestration for scalable and resilient database environments. EDB demonstrates its commitment to open-source innovation through contributions to the PostgreSQL community, advancing features such as incremental backups, JSON enhancements, and logical replication.

Running EDB on IBM Power servers provides benefits such as enhanced performance, security, and scalability. IBM Power10 processor technology optimizes complex query execution and supports large-scale data processing, making it suitable for enterprise workloads. Compatibility with Red Hat OpenShift on IBM Power enables organizations to deploy and manage containerized applications efficiently, using the robust IBM Power infrastructure for critical database operations.

For more information about running EDB on IBM Power, see the [IBM Power solution for EDB Postgres Advanced Server](#).

Frameworks

Frameworks are platforms that developers use to build software applications efficiently. They provide reusable code, libraries, tools, and best practices that simplify common tasks such as user interface design, database access, and system communication. This section describes some frameworks that are available on IBM Power.

.NET

.NET is a versatile framework that is developed by Microsoft for building applications such as web, desktop, mobile, gaming, and IoT. It supports multiple programming languages, including C#, F#, and Visual Basic, giving developers flexibility and choice. The framework includes a comprehensive class library and runtime environment that enable efficient application development and execution. .NET Core, a cross-platform version of .NET, enables developers to create applications that run on Windows, macOS, and Linux, extending its reach and usability.

One key feature of .NET is its support for modern development practices such as microservices architecture, cloud-native applications, and containerization. Integration with Azure, Microsoft's cloud platform, enables seamless deployment and scaling of applications in the cloud. The framework includes tools for automated testing, continuous integration, and continuous deployment or delivery (CD), supporting robust and efficient development workflows. Compatibility with development environments such as Visual Studio and Visual Studio Code enhances productivity and collaboration among developers.

The active .NET community and extensive documentation provide valuable resources for developers, fostering innovation and knowledge sharing. The framework is updated continuously with new features and improvements, helping ensure that it remains relevant and capable of meeting evolving development needs. Microsoft demonstrates its commitment to open-source development through .NET's open-source nature, which enables developers to contribute to its growth and use its capabilities for diverse projects. Overall, .NET is a robust and adaptable framework that enables developers to create high-quality applications across multiple platforms and industries.

.NET has growing support and options for running on IBM Power servers. This support enables organizations that use .NET technologies to use the reliability, performance, and security of the IBM Power Architecture. The primary way to run modern .NET applications on IBM Power is through Linux distributions that support the Power Little Endian (ppc64le) architecture, such as Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server, and Ubuntu. Microsoft actively supports .NET on these Linux distributions for the ppc64le architecture. In addition, Docker images for .NET applications are available for the ppc64le architecture, enabling containerized deployments on IBM Power.

Apache Kafka

Apache Kafka is a distributed streaming platform for building real-time data pipelines and streaming applications. Kafka operates as a publish-subscribe messaging system, enabling producers to publish streams of records to topics and consumers to subscribe to these topics to receive data. Topics are divided into partitions, which are ordered, immutable sequences of records. This partitioning enables horizontal scalability, enabling Kafka to handle large volumes of data by distributing it across multiple brokers (servers) in a cluster. Each partition is typically replicated across multiple brokers to help ensure fault tolerance and HA.

The Kafka architecture is built around several key components:

- ▶ *Brokers* are servers that host topic partitions. ZooKeeper manages the Kafka cluster by tracking the status of brokers and partitions.
- ▶ *Producers* write data to topics, selecting partitions (often based on a key).
- ▶ *Consumers* read data from topics and track their position (offset) within each partition.

This decoupled architecture enables producers and consumers to operate independently and at different rates. Kafka's design prioritizes durability and reliability, helping ensure that messages are persisted and delivered effectively, even during broker failures.

The versatility of Apache Kafka led to its widespread adoption across many industries. It is commonly used to build real-time data pipelines for analytics, data integration, and event-driven architectures. Use cases include tracking user activity on websites, processing financial transactions, ingesting sensor data from IoT devices, and powering real-time recommendation systems. Kafka's scalability, fault tolerance, and ability to handle high-throughput data streams make it a foundational technology for modern data infrastructure and real-time application development.

Security

This section describes some open-source security solutions that are available for IBM Power.

Keycloak

Keycloak is an open-source IAM tool that is developed by Red Hat. It provides authentication, authorization, and user management services for modern applications and services. Instead of building custom login and security logic, Keycloak offers a standardized, customizable, and secure solution.

Here are some key features of Keycloak:

- ▶ SSO
Users can log in once to access multiple applications without needing to authenticate again.
- ▶ Identity brokering and social login
Keycloak supports identity federation with external providers (like Google, Facebook, or corporate identity systems), enabling users to log in by using existing credentials.
- ▶ RBAC
Defines roles and permissions to control what users can access across applications and services.
- ▶ User federation
Integrates with existing LDAP or Active Directory systems to manage users centrally.
- ▶ MFA
Supports two-step authentication for enhanced security.
- ▶ Admin console and user self-service
Admins can manage users, roles, and settings through a web UI, and users can manage their own passwords and profiles.
- ▶ OAuth2, OIDC, and SAML Support
Keycloak supports all major authentication protocols, making it compatible with most modern applications and APIs.

OAuth

OAuth 2.0 is an authorization framework that separates the role of the client from the role of the resource owner. It uses authorization servers to issue access tokens to third-party clients with the explicit consent of the resource owner. These access tokens grant specific, limited access to resource servers that host the owner's protected resources.

The typical flow involves these actors:

1. Resource owner (user): The entity that owns the data.
2. Client application: The third-party application that requests access to the user's resources.
3. Authorization server: Issues access tokens after successfully authenticating the resource owner and obtaining their authorization.
4. Resource server: Hosts the protected user resources and enforces access control based on the access tokens that are presented by the client.

Different grant types define how the client obtains an access token (authorization code, implicit, or client credentials). The client first obtains an authorization grant with user consent, exchanges it for an access token at the authorization server, and then uses this token to make API requests to the resource server. OAuth 2.0 focuses on the delegation of authorization, not authentication, and uses bearer tokens to access resources.

Independent software vendors

An independent software vendor (ISV) is a company that creates, markets, and sells software applications that are designed to run on existing IBM platforms or OSs. IBM works closely with ISVs to optimize their solutions for IBM hybrid cloud and AI technologies. Through this collaboration, ISVs gain access to IBM global networks, technical resources, and go-to-market support. This partnership helps ISVs accelerate innovation and grow revenue while increasing the adoption of IBM products and platforms.

Security

This section introduces some ISV solutions for security on IBM Power.

CyberVR

CyberVR is a cybersecurity platform that uses virtual reality (VR) technology to enhance training and awareness. Developed by researchers at Sapienza University of Rome, CyberVR immerses users in interactive VR environments where they engage with realistic cybersecurity scenarios. This approach improves user understanding and retention of cybersecurity concepts by providing hands-on experience in a controlled, virtual setting. Studies show that CyberVR is as effective as traditional learning methods and more engaging, leading to better outcomes in cybersecurity education.

Beyond education, CyberVR serves as a tool for cybersecurity professionals to simulate and analyze potential threats in a virtual environment. By re-creating realistic attack scenarios, users can develop and refine response strategies without risking actual systems. This dual-purpose function makes CyberVR a valuable resource for both learning and practical application in cybersecurity.

CyberVR recently expanded its capabilities to support IBM Power servers, introducing the Isolated Recovery Environment (IRE) that is powered by Thin Digital Twin technology. This integration enables organizations to simulate and test cybersecurity scenarios on IBM Power platforms, including AIX, IBM i, Linux, and Windows, without affecting production systems. By creating high-fidelity, isolated replicas of entire IT environments, CyberVR enables teams to conduct testing, vulnerability assessments, and remediation drills in a secure, risk-free setting.

The IRE solution is optimized for IBM PowerVM, VMware, and x86 bare metal infrastructures, providing a scalable and automated approach to cyber resilience. This setup is valuable for industries with strict operational resilience requirements, such as those outlined by the Digital Operational Resilience Act (DORA). With CyberVR IRE, organizations can identify and address vulnerabilities proactively, helping ensure strong protection for critical workloads and compliance with regulatory standards.

Trend Vision One XDR security

Trend Vision One Security is a solution for IBM Power customers that protects across clouds, networks, devices, and endpoints with an AI-powered cybersecurity platform. With full support to run all components of the Trend Vision One platform on IBM Power, the solution provides administration and DevOps teams with greater control over their environment through centralized visibility and management. Using Trend Vision One on IBM Power helps organizations modernize, simplify, and converge security operations, enabling stronger protection against cyberthreats across diverse hybrid IT environments.

Trend Vision One delivers real-time insights on an executive dashboard, eliminating manual tasks and enabling efficient, informed decision-making. While IBM Power frees client resources to focus on strategic business outcomes, Trend Vision One automates cybersecurity reporting and playbooks for more efficient and productive security operations. Security teams can stay ahead of compliance requirements with real-time updates that help ensure the enterprise security posture remains strong.

For more information, see [Trend Vision One Endpoint Security](#).

Precisely Enforceive Enterprise Security Suite (IBM i)

The Precisely Enforceive Enterprise Security Suite is a comprehensive security and compliance solution for IBM i. It includes more than 20 fully integrated, GUI-controlled modules that enable system administrators and security officers to manage security and compliance tasks efficiently, even across multiple systems.

In an environment of privacy breaches, complex regulatory requirements, and evolving threats, the Enforceive Enterprise Security Suite strengthens IBM i defenses against unauthorized access. Its modules cover network security, authority swap, security monitoring, IBM i log transfer, and regulatory compliance. Other modules, such as Enforceive Field Encryption, Enforceive Password Self-Service, and Enforceive Firewall Manager, can be added to tailor the solution to specific requirements.

For more information, see [Enforceive Enterprise Security Suite](#).

Finance

This section describes financial solutions on IBM Power servers.

IBM Financial Transaction Manager

IBM Financial Transaction Manager (FTM) on IBM Power provides a robust solution for financial institutions that want to modernize their payment processing infrastructure. FTM integrates disparate payment systems, orchestrates complex transaction flows, and provides real-time monitoring and visibility across multiple payment types, including ACH, SEPA, SWIFT, and newer, faster payment schemes. By acting as a central hub, FTM simplifies the management of multiple payment channels and diverse data formats, creating a more streamlined and efficient operational environment.

Running FTM on an IBM Power infrastructure offers advantages. IBM Power servers deliver reliability, scalability, and strong security features, which are essential for financial transaction processing. FTM uses a modular, service-oriented architecture that enables institutions to adopt capabilities incrementally and integrate with existing systems. Key features include support for industry standards such as ISO 20022, real-time analytics and dashboards for operational insights, and the ability to process high transaction volumes with speed and precision. Deployments on IBM Power often use virtualization and containerization technologies such as Red Hat OpenShift, which enhance agility and flexibility in adapting to evolving business needs and regulatory requirements.

For more information about FTM, see [What is IBM Financial Transaction Manager?](#)

Temenos

Temenos Group AG is recognized as a leader in the 2022 Gartner Magic Quadrant for Global Retail Core Banking. Temenos has been an IBM Business Partner for more than 20 years and with Red Hat for more than 15 years to deliver its core banking solutions. Thousands of customers run traditional Temenos core banking on IBM Power.

In the second quarter of 2023, Temenos and IBM announced that the Temenos open platform was available on Red Hat OpenShift running on IBM Power (see [Temenos and IBM Help Banks Accelerate Their Core Banking Modernization with Hybrid Cloud](#)). This solution enables clients to use hybrid cloud benefits and accelerate digital transformation. By following the Temenos modernization path, banks can adopt a hybrid cloud strategy for core banking systems and use emerging technologies such as Explainable AI and digital banking. Currently, Temenos Transact R23 and R24 are certified to run on Red Hat OpenShift with IBM Linux on Power.

In10s Technologies

In10ns Technologies (Intense Technologies) offers banking solutions that enhance customer experience, streamline operations, and help ensure regulatory compliance. Its flagship product, UniServe NXT, is an AI-driven platform that supports customer communication management (CCM), data management, and process automation. The platform helps banks deliver personalized communications, automate financial reconciliations, and improve data accuracy. UniServe NXT integrates cloud technologies, including Red Hat OpenShift and IBM Power infrastructure, to deliver high performance and efficiency.

Fiserv

Fiserv is a global financial services technology company that supports more than 12,000 clients in over 80 countries and helps millions of consumers and businesses move and manage money efficiently. Fiserv offers a wide range of applications that are frequently deployed on IBM Power servers because of the platform's reliability, security, and performance. These applications cover core banking, payment processing, digital banking, risk and compliance, and data analytics.

In core banking, Fiserv platforms such as Signature and Premier are often deployed on IBM Power. Signature, particularly its international version, is being modernized to use IBM hybrid cloud capabilities, including Red Hat OpenShift on Power, to improve agility and scalability. Premier also benefits from the robust and scalable nature of IBM Power, enabling financial institutions to manage core banking operations efficiently and integrate add-on solutions.

GigaSpaces Digital Integration Hub

GigaSpaces Technologies Inc. is a privately held software company that was founded in 2000 and headquartered in New York City, with more offices in Europe, Asia, and Israel. The company provides HPC solutions that are focused on real-time analytics, distributed computing, and middleware technologies. GigaSpaces develops several products to address the evolving needs of modern enterprises.

One of its flagship offerings is the Smart Digital Integration Hub (DIH), a middleware solution that facilitates seamless software development and integration across diverse systems. GigaSpaces also offers GigaSpaces Cloud, a managed service on Google Cloud Platform (GCP), and AnalyticsXtreme, a real-time analytics platform built on a data grid architecture for processing and analyzing large data volumes. These innovations position GigaSpaces as a key player in distributed computing and cloud-native technologies.

GigaSpaces integrates its DIH solutions with IBM Power servers, enabling organizations to modernize IT infrastructures without full system migrations. This approach enables enterprises to use existing IBM i (AS/400), AIX, and Linux environments while adopting modern, cloud-native architectures. When deployed on Red Hat OpenShift clusters running on IBM Power, Smart DIH creates an event-driven, low-latency data fabric that decouples digital applications from core systems of record, accelerating digital transformation initiatives.

Also, GigaSpaces offers an enterprise retrieval-augmented generation (eRAG) platform developed in collaboration with IBM Watsonx.ai and IBM Watsonx Assistant™. This platform improves access to structured enterprise data through natural language queries, enabling intuitive interaction with complex datasets and enhancing decision-making. The combined capabilities of GigaSpaces DIH and eRAG, supported by IBM AI tools, provide a robust framework for organizations seeking to innovate and modernize operations on IBM Power servers.

Pipelines

This section describes some ISV-developed pipeline solutions that run on IBM Power.

GitLab Runner and Red Hat OpenShift Operator

GitLab Runner is an application that runs CI/CD jobs that are defined in GitLab pipelines. It runs on various platforms and is responsible for fetching code, running build and test scripts, and deploying applications.

On Red Hat OpenShift, GitLab Runner can be deployed as a containerized application, using Kubernetes-native orchestration. This approach enables CI/CD jobs to run efficiently within the same cluster where applications are deployed, enabling faster builds and better resource usage.

GitLab Runner is available as an operator for Red Hat OpenShift, which is designed to simplify the deployment and management of GitLab Runners. Operators automate common operational tasks by using custom resources.

Here are some key benefits:

- ▶ Automated lifecycle management: Install, upgrade, and scale GitLab Runners declaratively.
- ▶ Tighter GitLab integration: Register runners with GitLab by using credentials that are stored in Kubernetes secrets.

- ▶ Custom resource definitions (CRDs): Manage runner configurations as code, like other Kubernetes resources.
- ▶ Secure and isolated: Runs jobs within containers in Red Hat OpenShift, using security policies and namespaces.

Travis CI

Travis CI is a cloud-based continuous integration (CI) service that automates building, testing, and deploying software projects. It integrates with platforms such as GitHub, Bitbucket, and GitLab. Developers define build and test configurations in a `.travis.yml` file, which specifies the programming language, dependencies, and commands to run. When code changes are pushed to the repository, Travis CI automatically triggers the build and test processes and provides feedback on the results.

Key features of Travis CI include support for parallel and multi-environment builds, which enable faster testing and deployment. It integrates with notification systems such as email, Slack, and webhooks to keep developers informed about build status. Travis CI supports multiple CPU architectures, including IBM Power (ppc64le) and IBM Z (s390x), enabling developers to run CI/CD workflows on IBM Power servers. This capability helps ensure that applications are tested and deployed efficiently across environments. Travis CI can operate on-premises or in the cloud and is compatible with Red Hat OpenShift on IBM Power for deploying and managing containerized applications.

MuleSoft Anypoint Flex Gateway

To realize the full value of enterprise data, businesses often use APIs to improve products, operations, and systems, open new revenue streams, and provide insights that enhance business strategies and customer experiences.

Transporting data through APIs requires a protection layer to help ensure data security and restrict access to authorized actors. MuleSoft works with IBM to provide Anypoint Flex Gateway on IBM Power.

Empowering integration on IBM Power

IBM Power is known for its performance, reliability, and scalability. With the native integration of Anypoint Flex Gateway, businesses that use IBM Power servers can use the MuleSoft API management platform to connect diverse systems, applications, and data sources. Anypoint Flex Gateway is an Envoy-based, lightweight API gateway that is designed for high performance and enterprise-grade security. It integrates seamlessly with DevOps and CI/CD workflows, supporting demanding applications and microservices while helping ensure manageability across environments.

For more information, see [MuleSoft + IBM Power: Modernize and Manage Application Connectivity](#).

Backup and recovery

This section describes backup and recovery solutions that run on IBM Power.

Trilio Backup and Recovery

Trilio provides backup and recovery solutions for cloud-native environments, including Kubernetes and OpenStack. Designed for tenant-driven backup, disaster recovery (DR), and application mobility, Trilio helps ensure data protection and restoration across platforms. Its features include intelligent recovery, granular file and folder restoration, and automated backup processes to minimize downtime and maintain productivity.

For Kubernetes, Trilio offers scalable, agentless backup and DR, enabling rapid data recovery and compliance in regulated environments. The platform integrates with Red Hat OpenShift and other Kubernetes distributions, supporting migration and recovery across clusters. Trilio's patented Continuous Restore feature reduces recovery times, achieving near-zero Recovery Time Objectives (RTOs) and minimizing the financial impact of outages.

Trilio solutions are compatible with IBM Power servers, using the performance and scalability of IBM Power10 processor-based systems. This integration enables efficient deployment and management of containerized applications while helping ensure secure backup and recovery operations. By combining Trilio's data protection capabilities with IBM Power, enterprises can enhance resilience and maintain business continuity for critical workloads.

Veeam Backup and Replication

Veeam Backup and Replication is a data protection solution for virtual, physical, cloud, and SaaS environments. It creates image-level backups of VMs by using hypervisor snapshots for efficient data retrieval. For physical systems, Veeam uses agents to capture data. Backups can be full or incremental, storing only changed data blocks to optimize storage and speed up future backups. Supported methods include forward incremental, forever incremental, and reverse incremental, along with synthetic and active full backups to maintain backup chain health.

Veeam offers versatile recovery options to minimize downtime. Instant VM Recovery enables starting a VM directly from a backup file within minutes. File-level recovery restores specific files and folders from guest operating systems, and application-aware recovery helps ensure consistency for applications such as Microsoft Exchange, SQL Server, and Oracle. Veeam also supports VM replication for DR, creating ready-to-use copies of VMs onsite or offsite with configurable failover points.

Other features include data deduplication and compression to reduce storage and bandwidth usage, encryption for secure data transfer, and monitoring tools for real-time visibility into backup jobs and infrastructure health. Veeam supports immutable backups and integrates with secure storage options such as Veeam Vault to strengthen ransomware protection. Its scalability makes it suitable for environments ranging from small businesses to large enterprises, providing a comprehensive data protection strategy.

Veeam Backup and Replication and its associated components are fully supported on IBM Power servers, enabling enterprise-grade data protection for diverse workloads operating within this infrastructure. These integration capabilities are available for deployment in supported environments:

- ▶ **Veeam Agent for Linux on Power**

Veeam offers a Linux agent that is designed for IBM Power servers, supporting Linux distributions on IBM Power9 and Power10 architectures. The agent uses a “nosnap” package, which operates without the Veeam kernel module by using native file system snapshots. This capability enables backup and recovery of files and directories on Linux-based workloads running on IBM Power.

- ▶ **Veeam Agent for IBM AIX**

Veeam provides a dedicated agent for the IBM AIX OS, which is commonly used on IBM Power servers. This agent supports file-level backup of AIX LPARs and can be installed in the root file system or within a specific Workload Partition (WPAR). It integrates with Veeam Backup and Replication for centralized management and advanced recovery tasks. Supported AIX versions start from 6.1 Technology Level 5 (TL5).

- ▶ **Veeam Plug-ins for Enterprise Applications**

Veeam offers plug-ins for consistent backup and recovery of enterprise applications running on IBM Power:

- IBM Db2: Veeam provides a plug-in for IBM Db2 databases on Linux (including Linux on Power) and IBM AIX, helping ensure application-consistent backups.
- Oracle RMAN: A plug-in is available for Oracle databases on Linux, Oracle Solaris, and IBM AIX running on Power.
- SAP HANA: Veeam supports SAP HANA running on Linux on Power, enabling application-consistent backups.

- ▶ **Veeam Backup and Replication with IBM Cloud**

Veeam integrates with IBM Cloud to enable IBM Power users to use cloud resources for backup and DR. Veeam Backup and Replication can be deployed in IBM Cloud to protect on-premises Power workloads or workloads running natively in IBM Cloud. IBM and Veeam offer joint solutions for backup, migration, and DR on IBM Cloud, including support for hypervisor-level access for IBM Cloud VMware Solutions.

Veeam provides a suite of solutions to protect workloads running on IBM Power servers, including Linux, AIX, and key enterprise applications. It also offers integration with IBM Cloud and IBM Storage, enabling organizations to use Veeam’s backup and recovery capabilities for hybrid environments.

Precisely Assure MIMIX

Precisely is a software company that specializes in data integrity tools. The company is recognized for its expertise in helping organizations make better decisions based on trusted data. Assure MIMIX is an automated HADR solution that supports data replication and system recovery for AIX and IBM i.

Assure MIMIX for AIX

Assure MIMIX for AIX is a real-time, automated HA solution for IBM AIX servers that helps eliminate downtime and enables point-in-time data recovery. It protects against planned and unplanned outages and data loss.

The solution pairs real-time replication to a recovery server with continuous monitoring of cluster and application resources. If an outage occurs, applications fail over to the recovery server, which releases storage resources, manages IP addresses, reestablishes replication, mounts file systems, and restarts applications.

System availability extends beyond major disasters. Micro-disasters, such as database corruption or data loss that is caused by human error, can be equally costly. Affordable, manageable real-time DR protection for AIX applications and data is essential.

Assure MIMIX DR for AIX uses real-time replication and continuous data protection (CDP) technology to deliver near-instantaneous recovery from any point in time and supports rollback to recover from rolling disasters. It supports replication across any distance, between mixed storage and OS versions, and across physical, virtual, and cloud platforms. Assure MIMIX for AIX also integrates with IBM PowerHA for AIX to add point-in-time recovery and offsite protection against regional or site outages.

For more information, see [Assure MIMIX for AIX](#).

Assure MIMIX for IBM i

Assure MIMIX for IBM i is a HADR solution that is used by organizations of all sizes to prevent data loss and eliminate planned and unplanned downtime. It provides real-time, scalable replication with options for automated administration, monitoring and alerting, customizable switch automation, and a GUI. Assure MIMIX works across any combination of IBM i servers, storage, and operating system versions, supporting single-server environments and multi-site hybrid deployments.

Reliable DR requires maintaining an offsite replica of production data to help ensure resilience against site-level failures. Assure MIMIX uses IBM Remote Journaling technology for fast, efficient, and accurate replication across heterogeneous IBM i environments. Its Journal-Centric Data Group technology automates replication setup and management, even in large-scale environments with frequent journal creation.

Assure MIMIX includes features such as multi-threaded replication, browser-based and 5250 management, mobile monitoring, and guided configuration. These capabilities provide comprehensive automated protection for IBM i workloads, helping ensure business continuity and minimizing operational complexity.

For more information, see [Assure MIMIX](#).

Observability

This section describes observability solutions that are provided by ISVs for IBM Power.

Dynatrace OneAgent and Operator

Dynatrace is a monitoring platform that provides analytics and automation for unified observability and security. It supports use cases such as infrastructure observability, application observability, digital experience, log analytics, application security, threat observability, software delivery, and business analytics.

Dynatrace OneAgent discovers the processes running on the host. Based on its findings, OneAgent automatically activates instrumentation for the detected technology stack.

A Dynatrace ActiveGate acts as a secure proxy between Dynatrace OneAgents and Dynatrace clusters or between Dynatrace OneAgents and other ActiveGates that are closer to the Dynatrace cluster. In addition to routing monitoring data captured by OneAgents, Dynatrace ActiveGate can perform monitoring tasks by using APIs to query and monitor a wide range of technologies.

The Dynatrace and Red Hat certified operator is now supported on IBM Linux on Power with cloud-native, full-stack capabilities. The Dynatrace Operator manages the rollout and lifecycle of various Dynatrace components in Kubernetes and Red Hat OpenShift.

At launch, the Dynatrace Operator can deploy a containerized ActiveGate for Kubernetes API monitoring. Over time, new capabilities will be added to the Dynatrace Operator, including metric routing and API monitoring.

For more information and an overview of the Dynatrace monitoring stack, see [Dynatrace observability is now available for Red Hat OpenShift on the IBM Power Architecture](#).

For more information about the certified Red Hat OpenShift operator, see [Dynatrace Operator](#).

Splunk OpenTelemetry

Splunk OpenTelemetry (OTel) is an open-source observability framework that standardizes the collection of telemetry data across diverse environments. It provides tools, APIs, and software development kits (SDKs) to capture, generate, and export metrics, traces, and logs, enabling comprehensive analysis of software performance and behavior. By adopting OTEL, organizations can minimize vendor lock-in, gain deep insights into system performance, and enhance digital experiences.

The Splunk Distribution of the OpenTelemetry Collector is a specialized version that integrates with Splunk Observability Cloud. It enables the ingestion, processing, and export of telemetry data, providing a unified view of system health and performance. This distribution includes components from OpenTelemetry Core, OpenTelemetry Contrib, and other sources, ensuring robust data collection and improved support response from Splunk.

IBM collaborates with Splunk to support the OpenTelemetry Collector on IBM Power. This collaboration helps ensure that the collector operates efficiently on the IBM Power Architecture (ppc64le).

Crest Infosolutions Alfresco

The Alfresco Digital Business Platform is an open-source platform that helps organizations manage, govern, and use their content and processes effectively. It provides integrated services that enable businesses to build content-centric applications and streamline operations. The platform features a modular architecture, scalability, and flexibility, enabling it to adapt to various industry needs and deployment scenarios, including on-premises, cloud, and hybrid environments.

The Alfresco Digital Business Platform includes several key components:

- ▶ Alfresco Content Services provides enterprise-grade content management capabilities, including document management, version control, metadata management, collaboration tools, and robust security features.
- ▶ Alfresco Process Services, which is powered by Activiti, offers a BPM solution for automating workflows, managing tasks, and improving decision-making.
- ▶ Alfresco Governance Services helps organizations manage the information lifecycle, ensure compliance, and handle records management according to regulatory requirements. The platform enables businesses to unlock the value of their content, automate workflows, and drive digital transformation initiatives.

Crest Infosolutions is a global IT solutions and services provider with a strong focus on open-source technologies and is a prominent partner for the Alfresco Digital Business Platform. Crest offers comprehensive services around Alfresco, covering document management, workflow and BPM, and records management, helping organizations meet their digital transformation requirements.

Crest Infosolutions published a [white paper](#) that demonstrated a performance advantage when running Alfresco Content Services on IBM Power compared to an x86 solution.

Modernization by using cloud-native tools

This appendix describes how organizations can modernize IBM Power workloads by using cloud-native tools on Skytap on Azure. It explains how Azure-native services enable integration, improve scalability, and enhance automation for IBM i, AIX, and Linux on Power environments.

The following topics are described in this appendix:

- ▶ “Overview” on page 408
- ▶ “Skiytap on Azure” on page 408
- ▶ “Azure native tools for modernization” on page 411
- ▶ “AI with Azure” on page 426

Overview

Modernizing IBM i and AIX workloads requires a balanced approach that preserves the stability of existing systems while adopting cloud-native technologies. Skytap on Azure enables organizations to migrate and run IBM Power workloads, providing access to Azure-native tools for automation, DevOps, and AI-driven analytics.

Enterprise IT environments often include a mixture of IBM i, AIX, and x86 workloads. Traditionally, IBM i applications operated in isolated environments, limiting agility and integration with modern development frameworks. Skytap on Azure addresses this challenge by offering capabilities such as Live Clone for rapid environment duplication, integration with Azure DevOps for continuous integration and continuous delivery (CI/CD), and compatibility with Azure-native networking, storage, and AI services.

By using Azure OpenAI Service and Semantic Vector Search, enterprises can extract value from structured and unstructured IBM i data, making it accessible for generative AI-driven business intelligence and automation. Also, ARCAD's integration with Azure DevOps supports IBM i application development, enabling teams to implement version control, automated testing, and continuous deployment or delivery (CD) while maintaining compliance and security standards.

This section explains how Skytap on Azure, when combined with Azure-native tools, provides a scalable, cloud-integrated modernization approach for IBM Power workloads.

Skytap on Azure

The following sections provide an overview of Skytap on Azure, including its core capabilities, architecture, security features, and practical use cases.

What Skytap is

Skytap on Azure is a cloud-based infrastructure as a service (IaaS) platform that enables enterprises to migrate and run IBM Power servers, including IBM i, AIX, and Linux on Power, on Microsoft Azure. It provides a bare-metal IBM Power compute environment that preserves the operational consistency of on-premises Power servers while integrating Azure-native services for modernization. In May 2024, Kyndryl, an IT infrastructure services provider, acquired Skytap to strengthen its hybrid cloud services portfolio.

Note: Skytap is available on both Microsoft Azure and IBM Cloud platforms. To view the IBM Cloud regions where Skytap is available, see [Understanding regions](#). If your goal is to use Azure-native tools, select Skytap on Azure.

Architecture overview

Skytap on Azure supports the migration and operation of IBM Power and x86 workloads in the Microsoft Azure cloud. This integration enables organizations to use Azure services while maintaining the performance and reliability of existing applications.

Here are some key architectural components:

- ▶ **IBM Power servers integration**

Skytap on Azure provides native support for IBM Power workloads, including IBM Power9 and IBM Power10, helping ensure compatibility with enterprise applications running on IBM i, AIX, and Linux on Power. This support is delivered through dedicated IBM Power hardware that is hosted in Azure data centers, preserving the performance characteristics that are required for mission-critical applications.

- ▶ **Software-defined data centers**

The platform uses software-defined data center (SDDC) technology to replicate on-premises environments. This approach virtualizes infrastructure, storage, networking, operating systems (OSs), middleware, and applications, creating a comprehensive and flexible cloud environment.

- ▶ **Dynamic environment provisioning**

Skytap enables the creation and management of dynamic environments, enabling users to clone, deploy, and manage multiple instances of applications and systems. This capability is useful for development, testing, and training scenarios that require rapid provisioning and teardown of environments.

- ▶ **Networking and connectivity**

The architecture supports multiple networking configurations to help ensure secure and efficient connectivity:

- Azure ExpressRoute integration: Provides private, high-bandwidth connections between on-premises environments and Azure, reducing latency and enhancing security.

- Virtual Network (VNet) peering: Enables communication between Skytap environments and Azure services, supporting hybrid application architectures.

- ▶ **Storage architecture**

Skytap on Azure uses a scalable storage architecture for high availability (HA) and performance, which includes:

- Zettabyte File System (ZFS)-based storage nodes: Provide block-level storage with features such as data compression, snapshots, and cloning.

- Caching mechanisms: Multiple layers of caching, including client OS, Virtual I/O Server (VIOS), and storage node caching, optimize I/O performance.

- ▶ **Security and compliance**

The platform incorporates robust security measures to protect data and applications, including:

- Multi-layered security controls: Security is implemented at the physical, network, and application layers to safeguard against threats.

- Compliance certifications: Skytap on Azure meets industry standards, including SOC 1, SOC 2, SOC 3, PCI DSS, ISO 27001, and GDPR, helping ensure security and regulatory compliance.

Figure B-1 illustrates the architecture and connectivity model for Skytap on Azure. This hybrid design integrates on-premises data centers, Azure VNets, and Skytap environments running IBM Power and x86 workloads.

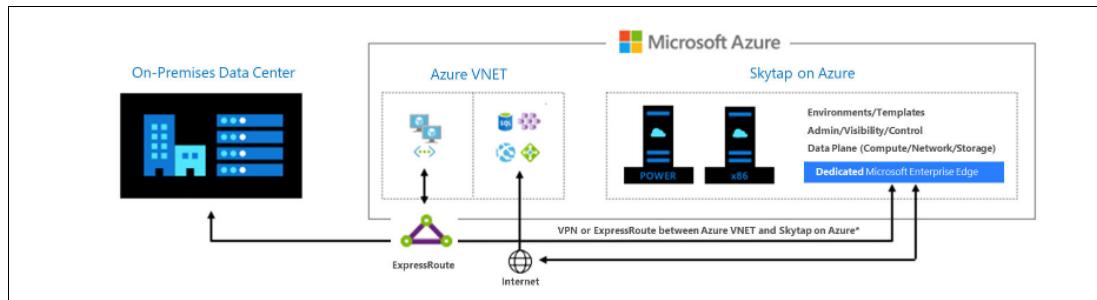


Figure B-1 Skytap on Azure Architecture

Organizations can securely connect Skytap workloads to on-premises networks by using Azure ExpressRoute, which provides low-latency, high-bandwidth connectivity. Also, virtual private network (VPN) or ExpressRoute connections between Azure VNets and Skytap enable access to Azure-native services, such as databases, application services, and monitoring tools. This integrated model supports hybrid cloud use cases and accelerates workload modernization without compromising security or performance.

Use cases

Skytap on Azure helps businesses migrate workloads to the cloud, modernize applications, and improve resilience.

Here are some key use cases:

- ▶ **Data center exit**
Migrate on-premises environments to Skytap with minimal disruption. Scale resources on demand, reduce capital expenditure (CapEx), and adopt a pay-as-you-go model.
- ▶ **Hardware refresh avoidance**
Run business-critical applications in Skytap without immediate hardware upgrades. Maintain stability while upgrading infrastructure at a controlled pace.
- ▶ **AI and application modernization with HA**
Automate provisioning, accelerate innovation, and integrate cloud-native services for enhanced scalability and reliability.
- ▶ **Disaster recovery (DR)**
Protect business operations by replicating on-premises environments in Skytap, helping ensure rapid recovery and minimal downtime.

Case studies

Organizations across various industries successfully use Skytap to modernize IT infrastructures, migrate critical workloads, and enhance DR capabilities. For more information, see these [win stories](#).

Azure native tools for modernization

Organizations modernizing IBM Power workloads on Skytap benefit from Azure-native services that enhance connectivity, storage, automation, and DevOps capabilities. These services enable integration between Skytap environments, on-premises infrastructure, and Azure resources, helping ensure operational efficiency and scalability.

This section provides an overview of key Azure solutions for networking, storage, DevOps, automation, and AI-driven modernization. It outlines best practices for integrating Azure ExpressRoute, VPN, and VNet peering for secure connectivity; Azure Blob Storage and Data Box Gateway for data management; and Azure DevOps with ARCAD for IBM i development workflows. It also explores automation with Terraform and Ansible, and the usage of Azure AI services to improve operational intelligence and workload performance.

By using these Azure-native tools, organizations can optimize IBM Power workloads on Skytap, improve reliability, reduce complexity, and accelerate their cloud transformation journey.

Networking

Skytap on Azure supports multiple networking options to help ensure secure, high-performance connectivity between Skytap environments, Azure-native services, and on-premises infrastructure. These networking solutions enable integration, optimize data transfer, and enhance workload performance in hybrid cloud deployments.

Figure B-2 shows the networking solutions in Skytap on Azure.

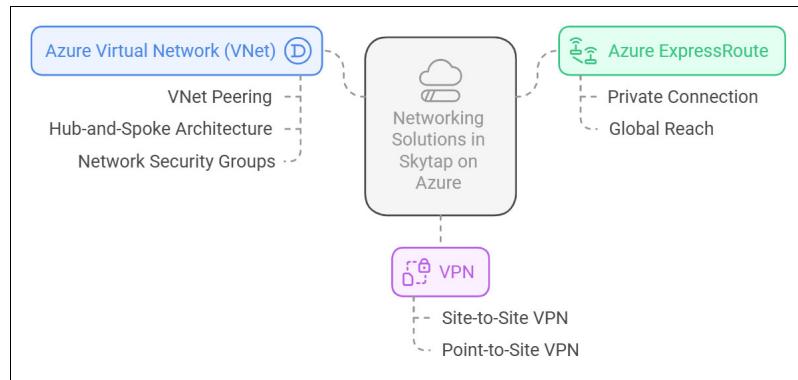


Figure B-2 Networking solutions in Skytap on Azure

Azure Express Route

Azure ExpressRoute provides a private, dedicated connection between on-premises environments, Skytap, and Azure, helping ensure low-latency, high-bandwidth connectivity for mission-critical workloads.

Azure ExpressRoute Global Reach extends this capability by enabling private connectivity across multiple ExpressRoute circuits. This feature facilitates secure data exchange between on-premises environments, Skytap, and Azure regions, enhancing hybrid cloud architectures and maintaining operational consistency.

Note: When you use ExpressRoute from Skytap to an Azure-native region, there are no ingress or egress charges for traffic between Skytap and Azure. However, customers are responsible for bandwidth and data egress costs when extending ExpressRoute to on-premises environments or other external networks. For more information about detailed pricing information, see [Explore pricing options](#).

VPN

Skytap supports multiple networking options to help ensure secure and efficient connectivity between on-premises environments, Skytap, and Azure-native services:

- ▶ VPN connectivity

Provides a secure and cost-effective option for integrating on-premises environments with Azure. VPN tunnels encrypt data in transit, helping ensure secure communication between Skytap environments and Azure services.

- Site-to-Site VPN

Establishes an IPsec-based encrypted tunnel between on-premises infrastructure and Skytap in Azure for secure workload connectivity.

- Point-to-Site VPN

Enables remote access for developers and administrators to connect securely to Skytap environments without requiring a full-site VPN.

- Azure VPN Gateway

Offers managed VPN services for integrating Skytap environments with Azure VNets, supporting hybrid cloud deployments.

- ▶ Azure VNet:

The VNet is the fundamental building block for private networking in Azure. VNets enable Skytap environments to communicate securely with Azure-native services, on-premises data centers, and other cloud resources.

- VNet peering

Directly connects Skytap environments to Azure VNets, enabling low-latency, high-throughput communication without requiring VPN or ExpressRoute.

- Hub-and-spoke architecture

Supports centralized network management by connecting Skytap to a central hub VNet, which acts as a gateway to other Azure resources.

- ▶ Network Security Groups

Apply security policies to control inbound and outbound traffic between Skytap, Azure services, and external networks.

Storage

Skytap on Azure integrates with Azure-native storage services for backup, archival, and DR. These services help optimize data retention, ensure compliance, and support workload recovery.

Azure Blob Storage

Azure Blob Storage is an object storage solution for scalable, secure, and cost-effective data retention. It supports multiple storage tiers, enabling organizations to balance performance and cost based on access frequency.

- ▶ Hot tier: Optimized for frequently accessed data.
- ▶ Cool tier: Cost-efficient for infrequently accessed data that still requires quick retrieval.
- ▶ Archive tier: For long-term retention. Requires rehydration before access.

For Skytap environments, Blob Storage is used for:

- ▶ Storing backups of IBM i, AIX, and Linux on Power workloads.
- ▶ Retaining snapshots for DR.
- ▶ Offloading data for regulatory compliance and long-term archival.

AzCopy for data migration

AzCopy is a CLI utility that transfers large datasets to and from Azure Blob Storage efficiently. It supports multi-threaded uploads, resumes interrupted transfers, and integrates with existing backup workflows.

- ▶ Enables bulk data migration from on-premises environments to Azure.
- ▶ Supports automated scheduling for routine backup transfers.
- ▶ Reduces downtime by optimizing large-scale data movement.

Azure Data Box Gateway

Azure Data Box Gateway is a network-based storage appliance that enables continuous data ingestion to Azure Blob Storage. It is useful for the following tasks:

- ▶ Transferring backup data from Skytap environments to Azure.
- ▶ Facilitating hybrid storage strategies for workloads that require ongoing data synchronization.
- ▶ Reducing latency in data transfers by using edge storage.

Backup and archival strategies

Organizations that use Skytap on Azure can implement various backup and retention strategies to help ensure data security and cost efficiency.

- ▶ Incremental backups store only the data that changed since the last backup. This approach reduces storage consumption and optimizes costs.
- ▶ Geo-Redundant Storage (GRS) replicates backup data to a secondary Azure region. This replication provides resilience against regional failures and helps ensure business continuity.

Policy-based retention automates data lifecycle management. It moves backups between hot, cool, and archive storage tiers based on access frequency. This process reduces long-term storage expenses while maintaining compliance.

Figure B-3 displays Skytap on Azure storage integration.

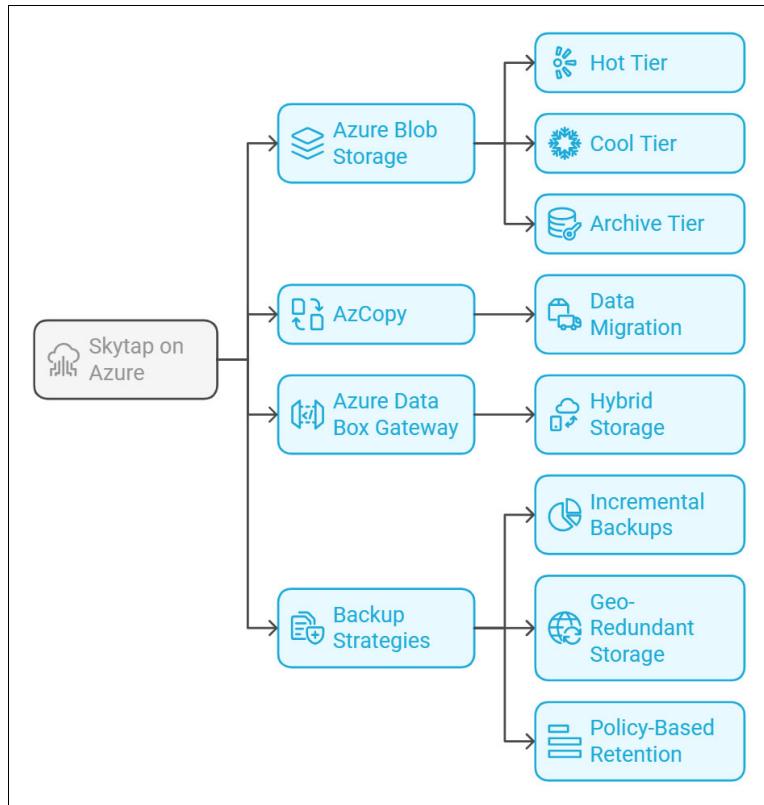


Figure B-3 Skytap on Azure storage integration

DevOps for IBM i

IBM i environments traditionally used stable but isolated development practices with manual source control, development, and deployment. As enterprises adopt DevOps, integrating IBM i into Git-based workflows, automated pipelines, and CI/CD processes is essential for agility and efficiency.

Microsoft Azure DevOps provides tools for source control, CI/CD, and project management, helping IBM i teams modernize. However, integrating IBM i into a cloud-based DevOps framework introduces cultural and technical challenges. Unlike open systems that rebuild and deploy full applications (such as .NET or Java), IBM i uses a delta build approach, compiling and deploying only modified objects.

Key considerations

Modernizing IBM i development with Azure DevOps enhances collaboration, automation, and efficiency. The following considerations outline essential aspects of this integration:

- ▶ Connecting IBM i and cloud-based DevOps teams requires a unified toolset. Microsoft Azure DevOps provides a platform for IBM i and .NET developers to share repositories, pipelines, and deployment strategies collaboratively.
- ▶ Using Git-based source control with Azure Repos enables IBM i developers to track versions, collaborate on code, and integrate with DevOps workflows. Developers can continue using IBM Rational Developer for i (IBM RDi), Visual Studio Code (VS Code), or 5250 green screens to manage code through Git.

- ▶ Automating deployments with Azure Pipelines streamlines CI/CD for IBM i applications, reduces manual tasks, and accelerates consistent releases.
- ▶ Managing IBM i workflows dynamically introduces branching strategies for project-based development. Unlike static branching in open systems, dynamic branching enables teams to isolate features, apply fixes, and manage parallel development efficiently.

Native cloud components

Using IBM i development with Azure DevOps improves the software development process. This combined platform supports planning, collaboration, and deployment. It simplifies tasks, enhances code quality, and accelerates delivery.

Figure B-4 shows the integration of IBM i with Azure DevOps.

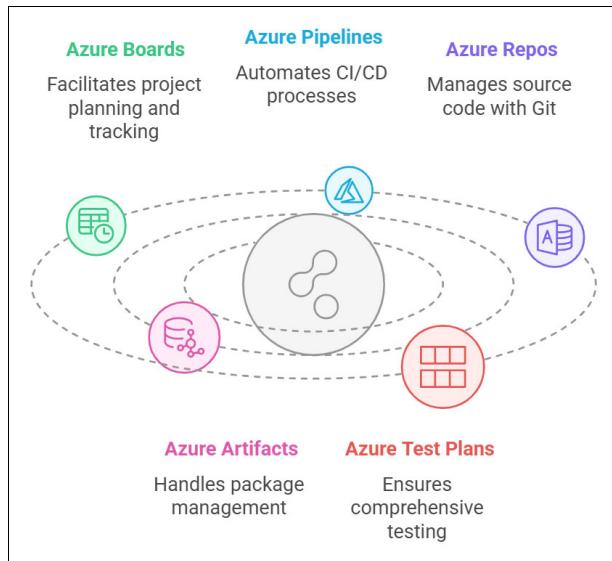


Figure B-4 Integrating IBM i with Azure DevOps

Here is a list of the Azure DevOps components:

- ▶ **Azure Boards**
Helps teams plan, track, and discuss work across projects. Its support for Kanban boards, backlogs, team dashboards, and custom reporting enables efficient project management. This visual approach improves collaboration among developers, testers, and project managers, helping ensure alignment throughout the development process.
- ▶ **Azure Pipelines**
Facilitates CI/CD for IBM i applications. By automating builds and deployments, it reduces manual errors, helps ensure consistent releases, and accelerates time-to-market. The pipeline's flexibility supports integration with various tools and services, meeting diverse development and deployment needs.
- ▶ **Azure Repos**
Manages source code by using Git repositories. This system provides version control, branching strategies, and pull requests (PRs), enabling collaborative development and code reviews. Teams can work concurrently on features and fixes, maintaining code integrity and facilitating seamless integration of changes.

- ▶ Azure Artifacts

Enables teams to create, host, and share packages across development environments. By integrating package management into CI/CD pipelines, it simplifies complex builds and helps ensure consistent access to required dependencies. This approach enhances build reliability and promotes code reuse.

- ▶ Azure Test Plans

Supports a comprehensive testing strategy that includes manual, exploratory, and automated tests. This approach helps ensure that IBM i applications meet quality standards before deployment. Integrated test reporting and analytics provide insights into test coverage and application quality, guiding continuous improvement efforts.

Note: Some IBM i customers have dependencies on x86 workloads, which include Windows and Linux environments^a. Because Microsoft Azure is a cloud platform for Windows and Linux, many organizations that use IBM i are likely to have an existing Azure subscription. This subscription provides a logical entry point for using Azure DevOps tools for IBM i modernization, integrating with existing development pipelines, and bridging IBM i with cloud-native workflows.

a. Source:

<https://community.ibm.com/community/user/discussion/the-11th-annual-ibm-i-marketplace-survey-results-are-ready-for-viewing>

ARCAD integration

Modernizing IBM i application development requires integrating traditional development workflows with cloud-based DevOps methodologies. ARCAD for DevOps provides IBM i teams with integration into Microsoft Azure DevOps, enabling automation, source control, and CI/CD workflows.

Azure DevOps offers a unified toolchain for managing development, testing, and deployment processes. However, native DevOps tools lack IBM i specific functions, such as dependency builds, impact analysis, and RPG-specific version control. ARCAD bridges this gap by enabling IBM i developers to work in VS Code, IBM RDi, and 5250 green screens while using Git-based repositories, automated builds, and deployment pipelines in Azure DevOps.

This section explains how ARCAD integrates with Azure DevOps and how it modernizes IBM i development by using cloud-native tools.

ARCAD architecture and tools

IBM i applications often operate within traditional environments, requiring structured modernization efforts. ARCAD integrates with Azure DevOps to automate source control, testing, and deployment workflows while maintaining platform integrity.

The following ARCAD tools enable structured modernization, CI/CD, and improved collaboration between IBM i and cloud development teams:

- ▶ **ARCAD Observer (Application Discovery)**

IBM i applications contain complex dependencies across RPG, COBOL, Control Language (CL), and Db2 for i. ARCAD Observer provides graphical insights into application architecture, database structures, and program calls. This visibility accelerates refactoring, modernization, and impact analysis for cloud migration or DevOps adoption.

- ▶ **ARCAD CodeChecker (code validation for RPG and COBOL)**

Maintaining code quality, security, and compliance is critical for modern IBM i applications. ARCAD CodeChecker automates static code analysis to detect inefficiencies, unused variables, performance bottlenecks, and security vulnerabilities. Integrated with Azure Pipelines, it helps ensure that code quality checks are embedded within CI/CD workflows.

- ▶ **ARCAD Builder (automated IBM i builds)**

IBM i uses a delta-based build approach, compiling only modified objects. ARCAD Builder automates dependency analysis, reducing manual compilation errors and accelerating build processes. When integrated with Azure Pipelines, it enables repeatable, automated builds while maintaining IBM i-specific logic.

- ▶ **ARCAD iUnit (automated unit testing)**

Unit testing is critical for modern DevOps pipelines. ARCAD iUnit enables automated RPG and COBOL unit tests, helping ensure that new code functions as expected. Integrated with Azure Test Plans, it facilitates continuous validation and prevents defects from moving into production environments.

- ▶ **ARCAD Verifier (regression testing for IBM i applications)**

Regression testing helps ensure that code updates do not introduce unintended issues. ARCAD Verifier captures baseline program behavior and automatically revalidates functions after code changes. Integrated with Azure DevOps, it automates regression testing within CI/CD workflows, reducing manual testing efforts and deployment risks.

- ▶ **DROPS for i (deployment and rollback management)**

IBM i applications require structured deployment processes to minimize downtime. DROPS for i enables automated, controlled releases across IBM i, Linux, Windows, and cloud environments. It supports rollback capabilities, helping ensure that failed deployments can be instantly restored, improving system resilience and operational stability.

Figure B-5 presents a summary of the ARCAD tools for IBM i modernization.

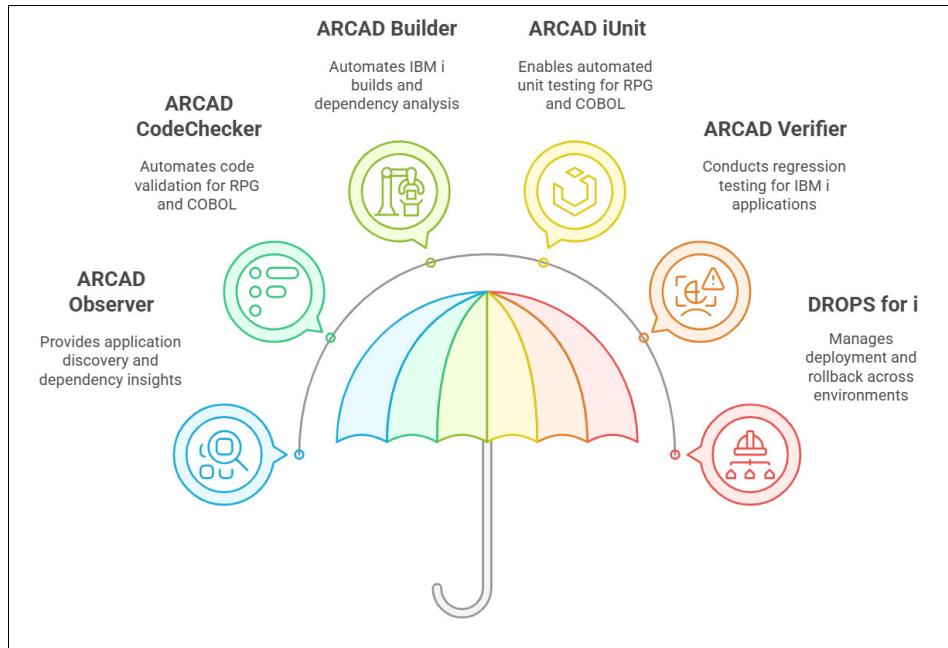


Figure B-5 ARCAD tools for IBM i modernization

Key components for Azure DevOps with ARCAD

The following components support IBM i modernization while maintaining compatibility with existing workloads:

► Azure Repos

Enables Git-based source control for IBM i applications, helping ensure version tracking, collaboration, and auditability. ARCAD integrates with Azure Repos, GitHub, and GitLab, allowing IBM i developers to:

- Manage RPG and COBOL code in Git while maintaining compatibility with IBM i object structures.
- Use VS Code, IBM RDi, or 5250 green screens for source code modifications.
- Apply branch-based development, enabling teams to work in parallel on multiple features or bug fixes.

► Azure Pipelines

Automates CI/CD for IBM i applications, reducing manual intervention and improving deployment efficiency. ARCAD integrates with Azure Pipelines to:

- Automate IBM i builds, helping ensure that all dependencies (logical files, service programs, physical files) are correctly compiled.
- Run impact analysis before each deployment to validate dependencies.
- Deploy across hybrid environments, supporting IBM i on Skytap, IBM PowerVS on IBM Cloud, or on-premises infrastructure.

► Azure Artifacts

Centralizes package management, enabling developers to distribute compiled IBM i components securely. ARCAD uses Azure Artifacts to:

- Store RPG, COBOL, and service program objects for controlled releases.
- Enable rollback capabilities, preserving previous versions for quick recovery.
- Manage application dependencies for IBM i microservices and hybrid cloud architectures.

► Azure Test Plans

Provides quality assurance and automated testing frameworks for IBM i workloads. ARCAD integrates with IBM i-specific testing tools, such as:

- iUnit for RPG and COBOL unit testing.
- ARCAD Verifier for regression testing, validating application behavior across IBM i versions.
- Automated User Acceptance Testing (UAT) execution by using Azure DevOps workflows.

AI assistant in ARCAD Discover

IBM i modernization is evolving with AI-driven automation, enhancing code analysis, impact assessment, and metadata management. ARCAD Discover, a key component of ARCAD 25, introduces an AI assistant that simplifies application understanding, functional mapping, and development workflows. This section explores the AI assistant's capabilities, integration within IBM i environments, and its impact on modernization efforts.

► Capabilities

The AI assistant provides the following capabilities for IBM i Application Discovery and management:

- Supports Application Discovery by identifying program dependencies, business rules, and data flows across IBM i environments.
- Automates impact analysis to detect relationships between components and predict potential issues before code modifications.
- Enables natural language queries for metadata searches, enabling developers to retrieve dependencies and application details efficiently.
- Generates structured views of IBM i applications by using functional tree optimization for improved navigation and maintainability.
- Provides AI-driven development insights to modernize IBM i codebases with minimal manual intervention.

► AI-driven metadata search

The AI assistant integrates natural language processing (NLP) to enable metadata queries. Developers and analysts can search for dependencies, logic flows, and system documentation by using structured or unstructured input.

► Automated code analysis and impact prediction

AI algorithms analyze dependencies within IBM i applications to help developers understand the impact of modifications:

- Predicts downstream effects of code changes and reduces regression risks.
- Automates dependency mapping to improve navigation across large codebases.

- ▶ Functional tree automation (planned feature)

AI-driven functional tree generation improves IBM i application structure by organizing code into modular, logical sections:

 - Reduces complexity by categorizing applications based on functional components.
 - Provides real-time insights into interactions between different program elements.
- ▶ Multi-engine AI for specialized tasks

The AI assistant uses multiple processing engines to optimize various tasks within IBM i modernization:

 - Code explanation: Interprets RPG, COBOL, CL, and SQL logic.
 - Application mapping: Generates diagrams and visual representations of data flow.
- ▶ AI and security

The ARCAD Discover AI assistant operates within secure environments, helping ensure that IBM i applications are analyzed without exposing source code externally. Key considerations include:

 - AI processing remains within enterprise-controlled environments to maintain security.
 - No requirement to access external AI models, helping ensure data privacy.
 - Deployment options include on-premises or private cloud instances based on compliance requirements.

Note: AI-driven insights that are generated by ARCAD Discover are processed locally or in controlled cloud environments, aligning with enterprise security policies.

- ▶ Business impact and use cases

AI-driven capabilities enhance efficiency, security, and modernization strategies across development, business, and compliance teams:

 - For developers and IT teams:
 - Automates code analysis to improve dependency detection and reduce manual review efforts.
 - Enhances modernization workflows by streamlining RPG code conversion.
 - For business analysts and project managers:
 - Generates AI-driven documentation, including metadata exports and application maps.
 - Provides structured insights that optimize decision-making for modernization strategies.
 - For security and compliance teams:
 - Detects security vulnerabilities in IBM i applications by using AI-based scanning techniques.
 - Helps ensure that AI processing remains within enterprise-controlled environments for compliance.

Monitoring and analytics

Skytap on Azure integrates with Azure-native monitoring tools to centralize system logs, provide real-time analytics, and improve operational visibility. These tools help you manage IBM Power workloads running on Skytap proactively and ensure compliance, performance optimization, and security.

Here are the Azure-native monitoring tools:

► **Azure Monitor**

Azure Monitor collects and analyzes telemetry data from Skytap environments to provide real-time insights into infrastructure health and performance. It enables you to:

- Track system logs and performance metrics to detect unusual activity in IBM i, AIX, and Linux environments running on Skytap.
- Set alerts and automated responses based on performance thresholds and anomaly detection.
- Integrate with Azure Log Analytics to improve log querying and long-term storage.

► **Azure Functions**

Azure Functions acts as a serverless automation layer that enables event-driven processing of Skytap logs. It supports:

- Automated log forwarding from Skytap to Azure Log Analytics and third-party SIEM tools.
- Execution of security rules and remediation actions in response to anomalies that are detected in log data.
- Integration with Azure Logic Apps to enable workflow automation that is based on Skytap audit logs.
- IBM i OS log forwarding to enable organizations to stream logs for centralized monitoring.

► **Azure Log Analytics**

Azure Log Analytics processes Skytap audit logs by using Kusto Query Language (KQL) for advanced log searches, correlation, and analysis. It enables you to:

- Investigate security events and operational issues by querying Skytap logs in real time.
- Generate compliance reports to meet regulatory requirements, such as SOC 1, SOC 2, and PCI DSS.
- Visualize Skytap user activities (for example, VM setup, deletion, and modification) by using structured reports.
- Support incident response by integrating log analytics with Azure Monitor.

► **Azure Dashboards**

Azure Dashboards provide customizable, interactive visualizations for monitoring Skytap environments. They offer:

- Real-time insights into log activity, including Skytap audit logs, infrastructure health, and security alerts.
- Integration with Log Analytics to display key performance indicators (KPIs) and system metrics.
- Notification capabilities to ensure that alerts are displayed and shared with IT Operations (ITOps) teams.

Note: Skytap audit logs integrate with Azure Webhooks and Logic Apps for ingestion and analysis:

- ▶ Audit Webhooks send logs to Azure Log Analytics by using OpenSSL authentication.
- ▶ Custom pipelines structure IBM Power workload logs for retention. Azure Functions.
- ▶ Azure Monitor alerts detect anomalies and security risks in real time.

Automation

Automation is essential for managing IBM Power workloads within Skytap on Azure because it improves operations, enhances scalability, and reduces manual effort. By using Azure-native automation tools, Ansible, and Terraform, you can deploy, configure, and manage workloads efficiently.

Automating operations

Skytap provides a REST application programming interface (API) that automates tasks such as VM provisioning, scaling, template creation, and storage management. The following sections describe different automation approaches.

Infrastructure as code with Terraform

Terraform is a declarative automation tool that provisions infrastructure in Skytap on Azure. You can define Skytap environments as code to help ensure consistency and repeatability. The following key use cases apply:

- ▶ Automating the deployment of IBM i, AIX, and Linux environments in Skytap.
- ▶ Defining networking, storage, and compute configurations in reusable Terraform modules.
- ▶ Managing ExpressRoute and VPN connectivity between Skytap and on-premises environments.

Example B-1 shows a Terraform configuration for provisioning a Skytap VM.

Example: B-1 Terraform configuration for Skytap VM

```
provider "skytap" {  
    username = var.username  
    api_token = var.api_token  
}  
  
resource "skytap_environment" "example" {  
    name          = "IBM i Test Environment"  
    description   = "Automated deployment of IBM i workloads"  
}  
  
resource "skytap_vm" "ibmi" {  
    environment_id = skytap_environment.example.id  
    template_id    = "12345"  
    vcpus          = 4  
    memory         = 16  
}
```

The Terraform configuration in Example B-1 on page 422 provisions a Skytap environment and deploys an IBM i VM through the following steps:

1. The Skytap provider connects by using API credentials.
2. A new environment is created with a specified name and description.
3. A VM instance is provisioned in this environment by using a predefined template ID.
4. The VM is allocated virtual CPUs (vCPUs) and memory according to workload requirements.

Note: For more information about infrastructure as code (IaC) with Terraform for Skytap, see [Using the Skytap Terraform provider](#). This resource includes details about the Terraform provider, getting started guides, use cases, and best practices for managing Skytap environments by using Terraform.

Automation with Ansible

Ansible provides a declarative, agentless approach to automating Skytap on Azure environments. By using Skytap's REST API, Ansible supports provisioning, configuration, and lifecycle management of IBM i, AIX, and Linux workloads.

The following key use cases apply:

- ▶ Managing Skytap VM power states.
- ▶ Scaling CPU and memory for IBM Power workloads.
- ▶ Managing storage, including logical unit number (LUN) operations.
- ▶ Automating template creation and deletion for workload replication.

The following examples show how to integrate Ansible with Skytap for automation. Controlling the power state of Skytap VMs helps ensure efficient resource usage and automation in cloud environments.

Example B-2 demonstrates how to start and stop Skytap VMs by using API-based automation.

Example: B-2 Starting and stopping a Skytap VM

```
- name: Start Skytap VM
  hosts: localhost
  tasks:
    - name: Power on VM
      uri:
        url: "https://cloud.skytap.com/vms/{{ vm_id }}/power"
        method: PUT
        headers:
          Content-Type: "application/json"
          Authorization: "Basic {{ skytap_auth }}"
        body: '{"runstate": "running"}'
        body_format: json

    - name: Stop Skytap VM
      hosts: localhost
      tasks:
        - name: Power off VM
          uri:
            url: "https://cloud.skytap.com/vms/{{ vm_id }}/power"
            method: PUT
            headers:
              Content-Type: "application/json"
```

```
Authorization: "Basic {{ skytap_auth }}"
body: '{"runstate": "stopped"}'
body_format: json
```

The playbook in Example B-2 on page 423 automates VM lifecycle operations to manage power states and integrate with broader automation workflows. The first task sends an API request to start the VM by setting its runstate to running. The second task stops the VM by updating the runstate to stopped. Authentication uses an API token in the request headers. This automation reduces manual effort and helps ensure that VMs can be controlled programmatically. Dynamically adjusting CPU and memory resources helps ensure optimal workload performance and cost efficiency.

Example B-3 shows VM specifications that are modified by using API requests.

Example: B-3 Scaling CPU and memory allocation

```
- name: Scale VM resources
  hosts: localhost
  tasks:
    - name: Modify CPU and memory allocation
      uri:
        url: "https://cloud.skytap.com/vms/{{ vm_id }}"
        method: PUT
        headers:
          Content-Type: "application/json"
          Authorization: "Basic {{ skytap_auth }}"
        body: '{"cpu": 8, "memory": 32}'
        body_format: json
```

The playbook in Example B-3 helps ensure that workloads can scale dynamically to meet demand without manual intervention. The API request updates CPU cores and memory allocation for a specified Skytap VM. The wanted specifications are defined in the body of the request. Authentication secures API calls to prevent unauthorized modifications. Automating resource scaling optimizes cost and performance across workloads. Skytap templates use workload replication to enable standardized deployments for development, testing, and DR.

Example B-4 creates a Skytap template by cloning an existing environment.

Example: B-4 Creating a Skytap template

```
- name: Create a new Skytap template
  hosts: localhost
  tasks:
    - name: Clone environment to create a template
      uri:
        url: "https://cloud.skytap.com/templates"
        method: POST
        headers:
          Content-Type: "application/json"
          Authorization: "Basic {{ skytap_auth }}"
        body: '{"source_environment_id": "{{ env_id }}", "name": "IBM i Template"}'
        body_format: json
```

The playbook in Example B-4 on page 424 automates template creation to reduce deployment time for recurring environments. The API request clones an existing environment and assigns it a template name. The template can be used for DR, testing, and workload replication. Versions and tags help track changes and optimize storage use. Automating template creation enhances agility and repeatability in cloud deployments. Automating network configurations helps ensure that Skytap workloads integrate with Azure VNets and hybrid architectures.

Example B-5 provisions a Skytap network and defines a subnet.

Example: B-5 Configuring a Skytap network

```
- name: Configure Skytap Network
  hosts: localhost
  tasks:
    - name: Create a new network
      uri:
        url: "https://cloud.skytap.com/networks"
        method: POST
        headers:
          Content-Type: "application/json"
          Authorization: "Basic {{ skytap_auth }}"
        body: '{"name": "Azure VNet", "subnet": "10.0.0.0/24"}'
        body_format: json
```

This playbook helps ensure efficient network configuration and integration with Azure services. The API request creates a network with a specified subnet. Azure VNet peering can extend Skytap connectivity across hybrid environments. Security controls, including firewall rules and access lists, should be applied after deployment. Automating network configurations improves scalability and deployment consistency.

Automating with Azure Pipelines

Users can automate the deployment, configuration, and management of IBM i, AIX, and Linux workloads by using Azure DevOps Pipelines. Integrating Terraform for provisioning and Ansible for configuration helps ensure a repeatable and secure automation framework.

- ▶ Infrastructure provisioning with Terraform

Azure Pipelines runs Terraform configurations to provision and manage Skytap environments, including VMs, networks, and templates. By using the Skytap Terraform provider, teams can define IaC and enforce consistency across deployments.

- ▶ Configuration management with Ansible

After Terraform provisions Skytap VMs, Azure Pipelines runs Ansible playbooks to automate configuration, software installation, security settings, and workload deployments. This process helps ensure that IBM i, AIX, and Linux instances in Skytap align with enterprise policies and best practices.

- ▶ Workflow integration in Azure Pipelines

The following stages outline the automation process:

- a. Terraform deployment stage

- Initializes Terraform and applies configurations for Skytap resources.
- Provisions IBM i, AIX, or Linux VMs in Skytap with predefined specifications.

- b. Ansible configuration stage
 - Runs Ansible playbooks to apply system settings and software configurations.
 - Helps ensure consistency and compliance across Skytap workloads.
- c. Monitoring and logging
 - Uses Azure Monitor and Azure Log Analytics to track Terraform and Ansible execution.
 - Provides real-time visibility into infrastructure state and performance.

AI with Azure

Enterprises gain a competitive advantage by integrating IBM i, AIX, and Linux on IBM Power servers with cloud-based AI. With Skytap on Azure, you can migrate mission-critical workloads to the cloud and use Azure AI services for advanced analytics, automation, and insight generation. This approach accelerates generative AI initiatives and enables real-time, data-driven solutions that were previously difficult because of earlier constraints.

Modernizing data for generative AI

Generative AI systems rely on large volumes of unstructured and structured data to train models and generate responses. By migrating IBM Power workloads to Skytap on Azure, enterprises can securely move data closer to Azure AI services, including Azure OpenAI and Semantic Vector Search, while preserving compatibility with earlier applications. This approach supports retrieval-augmented generation (RAG), which enables generative AI queries to reference enterprise-specific data in real time.

Key steps in generative AI adoption with Skytap on Azure include:

- ▶ Cloud migration
 - Lift and shift IBM Power workloads to Skytap to reduce physical infrastructure.
 - Integrate with Azure ExpressRoute or VPN for low-latency and secure connectivity.
- ▶ Data preparation
 - Label, format, or anonymize data for generative AI training.
 - Store unstructured content (such as documents and logs) in Azure Data Lake or vector databases for embedded retrieval.
- ▶ Model training and fine-tuning
 - Use Azure OpenAI or base large language models (LLMs) to train or fine-tune enterprise-specific models.
 - Help ensure data privacy by using secure cloud architecture and Skytap native audit logs.
- ▶ Chat interface and RAG
 - Deliver generative AI responses by using Azure OpenAI chat completion.
 - Deploy semantic search to enhance user queries with enterprise data from Skytap environments.

Skytap AI Assistant

Skytap AI Assistant uses generative artificial intelligence (AI) to optimize resource management, provide contextual insights, and streamline operations. It operates in two knowledge domains:

- ▶ Skytap-specific: Answers queries about Skytap documentation, user guides, and best practices.
- ▶ General knowledge: Provides broad technical information, including industry standards and AI fundamentals.

By integrating Azure serverless services, AI Assistant can scale dynamically and deliver real-time recommendations on system performance, user workloads, and troubleshooting. This capability supports both technical teams, which benefit from in-depth Skytap knowledge, and non-technical teams, which require intuitive cloud operations guidance.

Benefits of AI

Here are the benefits of AI in Skytap on Azure:

- ▶ Data proximity: Keeps enterprise data close to Azure AI for efficient model training and real-time inference.
- ▶ Reduced complexity: Eliminates the need to rewrite IBM i or AIX workloads so that organizations can focus on cloud-enabled analytics.
- ▶ Enhanced security: Maintains data governance by using Skytap native auditing and Azure compliance frameworks.
- ▶ Scalable compute: Uses Azure high-performance infrastructure for generative AI to help ensure on-demand resource availability.

Note: By combining Skytap on Azure with Azure AI services and generative AI models, you can achieve new levels of operational efficiency, market responsiveness, and innovation while preserving the reliability and performance of systems.



C

Details for performance claims

This appendix provides the technical details behind the performance claims that are made in the book.

The following topics are described in this appendix:

- ▶ “Improved performance” on page 430
- ▶ “Running AI on a highly performant sustainable platform” on page 430
- ▶ “Improved economics” on page 430

Improved performance

This section describes the supporting details for the improved performance of artificial intelligence (AI) on IBM Power that is described in “Improved performance” on page 47.

These details are derived from IBM internal testing of question-and-answer inferencing by using the PrimeQA model¹ (which is based on Dr. Decr and ColBERT models). The 42% improvement is based on total throughput in scores (inferences) per second on an IBM Power S1022 server (one 20-core and 512 GB) running SMT-4, compared to an Intel Xeon Platinum 8468V-based system (One 48-core and 512 GB).

Table C-1 shows the results.

Table C-1 Test results

Server	Inferences per second	Concurrent Users
Power S1022	6.26	40
Supermicro SYS-221H-TNR ^a	4.4	40

a. For more information, see [Hyper SuperServer SYS-221H-TNR](#).

The following information applies to these results:

- ▶ The results are valid as of August 22, 2023, and were obtained under laboratory conditions. Individual results can vary based on workload size, usage of storage subsystems, and other factors.
- ▶ The Power S1022 server (two 20-core, 2.9 - 4 GHz, and 512 GB) used a NUMA-aligned 10-core logical partition (LPAR). The test was run with Python and Anaconda environments, including Python 3.9 and PyTorch 2.0 packages. The Python libraries were platform-optimized for both Power and Intel systems.
- ▶ The batch size was 60 with 40 concurrent users. The `torch.set_num_threads(int)` parameter was optimized across various load levels. The models were fine-tuned by IBM on a corpus of IBM-internal data.

Running AI on a highly performant sustainable platform

This section describes the supporting details for the improved performance of AI on IBM Power that is described in “Running AI on a highly performant sustainable platform” on page 48.

These details are derived from IBM internal testing of data science components (WML, Windows Subsystem for Linux (WSL), and Analytic Engine) of IBM Cloud Pak for Data (CP4D) 4.8 running on Red Hat OpenShift 4.12. The results are valid as of November 17, 2023, and were obtained under laboratory conditions. Individual results can vary based on workload size, use of storage subsystems, and other factors.

Improved economics

This section describes the supporting details for the improved performance of AI on IBM Power that is described in “Improved economics” on page 48.

¹ Source: <https://github.com/primeqa>

These details are derived from on IBM internal testing of data science components (WML, Windows Subsystem for Linux (WSL), and Analytic Engine) of IBM Cloud Pak for Data (CP4D) 4.8 running on Red Hat OpenShift 4.12. The results are valid as of November 17, 2023, and were obtained under laboratory conditions. Individual results can vary based on workload size, usage of storage subsystems, and other factors.

The workload simulates a real-time fraud detection logic flow. JMeter submits credit card transactions for different user ID and card number combinations. The inferencing application, running as microservices in the CP4D deployment space, extracts the user ID and credit card number and uses them to look up six previous transactions of the same user and card combination from the Db2 database, which also runs within the CP4D cluster. The data that is retrieved from the database is combined with the new entry and passed to the LSTM model to determine whether the latest transaction is fraudulent. The score (a value 0 - 1) is returned to the JMeter client as an indicator of whether the transaction is likely fraudulent.

The measurement for both Power server and Intel servers is the throughput result (scores per second) that is reported by JMeter when running 192 concurrent threads (one thread representing one user) against 96 inferencing endpoints.

Here are the system configuration details for IBM Power10 S1022 and Intel Xeon Platinum 8468V:

- ▶ IBM Power10 S1022
 - 40 physical cores and 2 TB RAM (machine type 9105-22A)
 - Seven LPARs:
 - Three master nodes: Two cores with 32 GB RAM each
 - Three worker nodes: Ten cores with 490 GB RAM each
 - One bastion node: Four cores with 128 GB RAM
 - Local 800 GB NVMe drives for start, and one 1.6 TB NVMe for NFS storage on the bastion node
 - One 100 Gb Ethernet adapter virtualized through SR-IOV (each LPAR uses 10% bandwidth)
 - CPU frequency: 3.20 GHz - 4.0 GHz
 - Worker nodes: SMT-4
 - Master and bastion nodes: SMT-8
- ▶ Intel Xeon Platinum 8468V
 - 96 physical cores and 2 TB RAM
 - KVM host: 2 cores and 32 GB RAM
 - Seven KVM guests:
 - Three master nodes: Four cores with 32 GB RAM each
 - Three worker nodes: Twenty-four cores with 490 GB RAM each
 - One bastion node: Four cores with 128 GB RAM
 - Local 1.6 TB NVMe drives for start, and one 1.6 TB NVMe for NFS storage on the bastion node
 - One 100 Gb Ethernet adapter virtualized through SR-IOV
 - CPU frequency: 2.40 GHz - 3.8 GHz
 - All nodes: RHEL CoreOS KVM guests with hyperthreading enabled

Pricing is based on IBM Power S1022², typical industry-standard Intel x86 pricing³, and IBM software pricing⁴.

² Source: <https://www.ibm.com/power/pricing/us-en#oracle>

³ Source: <https://www.synnexcorp.com/us/govsolv/pricing/>

⁴ Source: <https://www.ibm.com/downloads/cas/DLBOWBPK>

Abbreviations and acronyms

ACL	Access Control List	CP4I	IBM Cloud Pak for Integration
ACS	Access for Client Solutions	CPN	compute plane node
ADDS	Active Directory Domain Services	CRB	CodeReady Builder
ADFS	Active Directory Federation Services	CRD	custom resource definition
AES	Advanced Encryption Standard	CRIU	Checkpoint/Restore in Userspace
AFM	Active File Management	CRM	Customer Relationship Management
AI	artificial intelligence	CSI	Container Storage Interface
AIO	All-in-One	CTE	Common Table Expression
AMS	Active Memory Sharing	CVE	Common Vulnerability Exposure
AOT	Ahead-of-Time	DAST	Dynamic Application Security Testing
API	application programming interface	DBMS	database management system
ARM	Application Resource Management	DCM	dual chip module
AWS	Amazon Web Services	DDD	domain-driven design
BPM	Business Process Management	DIH	Digital Integration Hub
BRMS	Backup Recovery and Media Services	DLPAR	dynamic logical partition or partitioning
CAGR	compound annual growth rate	DMD	Domain Model Definition
CapEx	capital expenditure	DORA	Digital Operational Resilience Act
CCM	customer communication management	DR	disaster recovery
CD	continuous deployment or delivery	ECC	error-correcting code
CDP	continuous data protection	EDB	EnterpriseDB
CEAC	Common Europe Advisory Council	EDR	Endpoint Detection and Response
CI/CD	continuous integration and continuous deployment or delivery	EEI	Execution Environment Image
CI	continuous integration	EFS	Elastic File Store
CIFS	Common Internet File System	ESS	Elastic Storage System
CIO	Chief Information Officer	EUM	end-user experience monitoring
CL	Command Language or Control Language	FC	Fibre Channel
CLI	command line interface	FFDC	First Failure Data Capture
CLI	command-line interface	FHE	Fully Homomorphic Encryption
CMC	Cloud Management Console	FIM	federated identity management
CNCF	Cloud Native Computing Foundation	GA	generally available
CNFS	Clustered Network File System	GCP	Google Cloud Platform
CoD	Capacity on Demand	HA	high availability
CP4A	IBM Cloud Pak for Automation	HADR	high availability and disaster recovery
CP4BA	IBM Cloud Pak for Business Automation	HCI	Hyper-converged Infrastructure
CP4D	IBM Cloud Pak for Data	HMC	Hardware Management Console
		HNV	Hybrid Network Virtualization
		HPC	high-performance computing

IaaS	infrastructure as a service	NVMe	Nonvolatile Memory Express
IaC	infrastructure as code	OAuth	Open Authorization
IAM	Identity and Access Management	ODM	Object Data Manager
IASP	independent auxiliary storage pool	OIDC	OpenID Connect
IBM	International Business Machines Corporation	OLTP	online transaction processing
IDE	integrated development environment	OS	operating system
IFS	Integrated File System	OSS	open source software
IKEv2	Internet Key Exchange version 2	OTel	OpenTelemetry
ILE	IBM Integrated Language Environment	PCI	Peripheral Component Interconnect
IoT	Internet of Things	PEP2	Power Enterprise Pools 2.0
IPL	initial program load	PFA	Predictive Failure Analysis
ISA	instruction set architecture	PHYP	PowerVM Hypervisor
ISV	independent software vendor	POM	Project Object Model
ITOps	IT Operations	PR	pull request
JIT	Just-In-Time	PV	persistent volume
K8s	Kubernetes	PVC	persistent volume claim
KPI	key performance indicator	QEMU	Quick Emulator
KVM	Kernel-based Virtual Machine	QoS	quality of service
LDAP	lightweight directory access protocol	QSE	quantum-safe encryption
LKU	Live Kernel Update	Qshell	QSH
LLM	large language model	RAC	Real Application Cluster
LLU	Live Library Update	RAG	retrieval augmented generation
LP	logical port	RAS	reliability, availability, and serviceability
LPAR	logical partition	RBAC	role-based access control
LPM	Live Partition Mobility	RCS	Right Computer Systems
LUN	Logical Unit Number	RDBMS	relational database management system
LV	logical volume	IBM RDi	IBM Rational Developer for i
Merlin	IBM i Modernization Engine for Lifecycle Integration	REST	Representational State Transfer
MFA	multi-factor authentication	RHEL	Red Hat Enterprise Linux
ML	machine learning	ROI	return on investment
MMA	Matrix Math Accelerator	ROX	ReadOnlyMany
MTTR	mean time to resolution	RPA	Robotic Process Automation
NAS	network-attached storage	RPO	Recovery Point Objective
NAT-T	Network Address Translation-Traversal	RSCT	Reliable Scalable Cluster Technology
NFS	Network File System	RTO	Recovery Time Objective
NIC	Network Interface Controller	RWO	ReadWriteOnce
NIM	Network Installation Manager	RWOP	ReadWriteOncePod
NLP	natural language processing	RWX	ReadWriteMany
NPIV	N_Port ID Virtualization	SaaS	software as a service
NPM	Node Package Manager	SAML	Security Assertion Markup Language
		SAN	storage area network

SBM	Saudi Business Machines
SCM	single-chip module
SDDC	software-defined data center
SDS	software-defined storage
SEA	Shared Ethernet Adapter
SLA	service-level agreement
SME	subject matter expert
SMS	System Management Services
SPP	shared processor pool
SRE	Site Reliability Engineer
SRR	Simplified Remote Restart
SSD	solid-state drive
SSH	Secure Shell
SSO	single sign-on
SSP	Shared Storage Pool
TCB	Trusted Computing Base
TCO	total cost of ownership
TE	Trusted Execution
TL5	Technology Level 5
TOGAF	The Open Group Architecture Framework
TSD	Trusted Signature Database
UDF	user-defined function
UI	user interface
VEB	Virtual Ethernet Bridge
VEPA	Virtual Ethernet Port Aggregator
VF	Virtual Function
VIOS	Virtual I/O Server
VLAN	virtual local area network
VM	virtual machine
VMRM	IBM Virtual Machine Recovery Manager
VNet	Virtual Network
vNIC	virtual Network Interface Controller
VPN	Virtual Private Network
VR	virtual reality
VS Code	Visual Studio Code
vSCSI	virtual SCSI
WPAR	Workload Partition
WSL	Windows Subsystem for Linux
ZFS	Zettabyte File System
ZTA	Zero Trust Architecture

Related publications

The publications that are listed in this section are considered suitable for a more detailed description of the topics that are covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topics in this document. Some publications that are referenced in this list might be available in softcopy only.

- ▶ *IBM Power10 Scale-Out Servers Technical Overview S1012, S1014, S1022s, S1022 and S1024*, REDP-5675
- ▶ *IBM Power E1050 Technical Overview and Introduction*, REDP-5684
- ▶ *IBM Power E1080 Technical Overview and Introduction*, REDP-5649
- ▶ *IBM PowerHA SystemMirror for AIX Cookbook*, SG24-7739
- ▶ *IBM PowerHA SystemMirror and IBM VM Recovery Manager Solutions Updates*, REDP-5694
- ▶ *IBM Power Systems SR-IOV: Technical Overview and Introduction*, REDP-5065
- ▶ *IBM Storage Ceph Concepts and Architecture Guide*, REDP-5721
- ▶ *IBM Storage Fusion Product Guide*, REDP-5688
- ▶ *IBM Storage Scale Information Lifecycle Management Policies*, REDP-5739
- ▶ *IBM Storage Scale System Introduction Guide*, REDP-5729
- ▶ *Introduction to IBM PowerVM*, SG24-8535
- ▶ *Simplify Management of IT Security and Compliance with IBM PowerSC in Cloud and Virtualized Environments*, SG24-8082
- ▶ *Using Ansible for Automation in IBM Power Environments*, SG24-8551
- ▶ *Using Pacemaker to Create Highly Available Linux Solutions on IBM Power*, SG24-8557

You can search for, view, download, or order these documents and other Redbooks, Redpapers, web docs, drafts, and additional materials, at the following website

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources

- ▶ NIC - Introducing New PowerVM Virtual Networking Technology blog
<https://community.ibm.com/community/user/power/blogs/charlesgraham1/2020/06/19/vnic-introducing-a-new-powervm-virtual-networking?CommunityKey=71e6bb8a-5b34-44da-be8b277834a183b0&tab=recentcommunityblogsdashboard>
- ▶ What is Docker?
<https://www.ibm.com/think/topics/docker>

- ▶ What is Kubernetes?
<https://www.ibm.com/think/topics/kubernetes>
- ▶ What is Podman?
<https://www.redhat.com/en/topics/containers/what-is-podman>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



Redbooks

Modernization Techniques for IBM Power

SG24-8582-00

ISBN 0738462179



(1.0" spine)
0.875" <-> 1.498"
460 <-> 788 pages



SG24-8582-00

ISBN 0738462179

Printed in U.S.A.

Get connected

