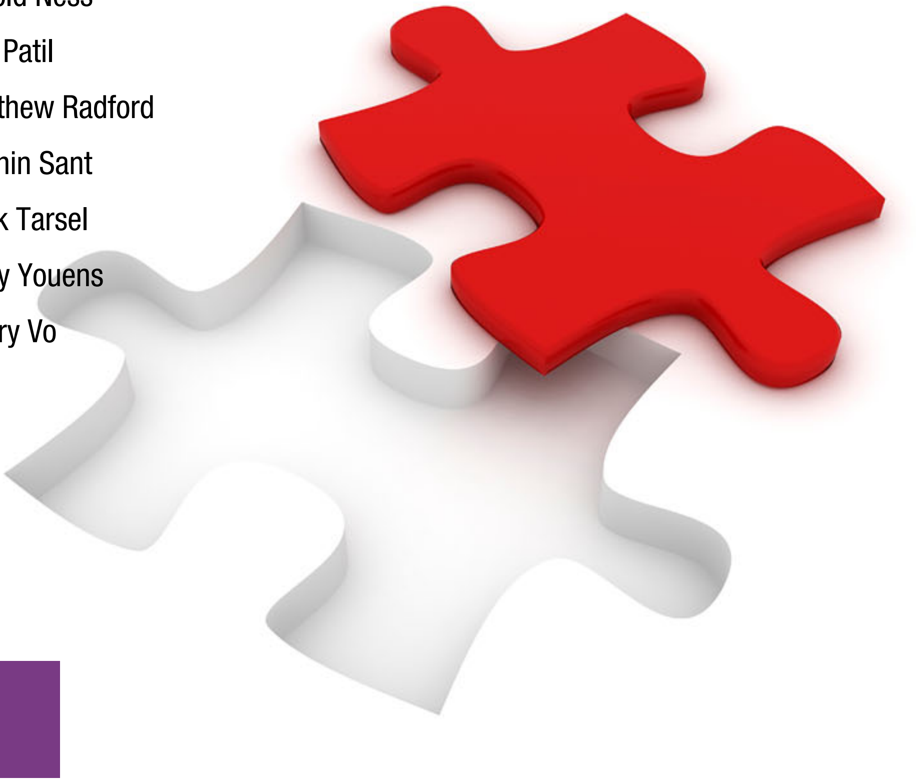
Redbooks
ibm.com/redbooks

Modernization Techniques for IBM Power

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



IBM Power

Artificial Intelligence

IBM

Redbooks

IBM Redbooks

Modernization on Power

May 2025

SG24-8582-00

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

First Edition (May 2025)

This edition applies to:
AIX Version 7.2 and 7.3
IBM i Version 7.4, 7.5 and 7.6.
IBM Power10 based processors
Red Hat OpenShift 4.17 and 4.18
PowerVM 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

© Copyright IBM Corp. 2025.

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.

© Copyright IBM Corp. 2025.

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®
C®
Code Assistant™
Concert®
Db2®
DB2®
DS8000®
FlashCopy®
GDPS®
Global Business Services®
Granite®
HyperSwap®
IBM®
IBM API Connect®
IBM Cloud®
IBM Cloud Pak®
IBM Consulting™
IBM FlashSystem®
IBM Instana™
IBM Partner Plus®
IBM Security®
IBM Services®
IBM Spectrum®
IBM Watson®
IBM watsonx®
IBM Z®
Insight®
Instana®
Integrated Language Environment®
Interconnect®
Language Environment®
Micro-Partitioning®
Netcool®
Orchestrate®
Parallel Sysplex®
POWER®
Power Architecture®
Power8®
Power9®
PowerHA®
PowerVM®
pureScale®
Rational®
Redbooks®
Redbooks (logo) ®
Spyre™
System z®
SystemMirror®
Turbonomic®
Watson Analytics®

watsonx®
watsonx Assistant™
watsonx Code Assistant™
watsonx Orchestrate™
watsonx.ai®
watsonx.data®
watsonx.governance®
WebSphere®
X-Force®
z/OS®
z/VM®

The following terms are trademarks of other companies:

Evolution, are trademarks or registered trademarks of Kenexa, an IBM Company.

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.

Ansible, Ceph, CloudForms, Fedora, JBoss, OpenShift, Red Hat, RHCA, 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

The rapid global changes of recent years have significantly impacted IT. Organizations are accelerating their digital transformations to support an increasingly always-on world. IT leaders are focused on modernizing critical processes and operations to gain a competitive edge in this dynamic environment.

Application modernization—updating applications for improved maintenance, extension, deployment, and management—is key to meeting current and future needs. It offers numerous business and technical advantages.

Modernizing existing enterprise applications facilitates a smoother transition to hybrid cloud, providing the flexibility to run applications anywhere, anytime. A cloud-native microservices approach maximizes the scalability and agility of the cloud.

Modernizing on IBM Power allows new cloud-native microservices to integrate with existing applications, leveraging the platform's performance, reliability, and security. This removes integration and productivity barriers, paving the way for enhanced user experiences, new applications, and ultimately, new business opportunities.

This IBM Redbook offers 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 in today's world. The content is designed 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 in Tulsa, Oklahoma, USA. He has over 40 years of experience with IBM primarily in a technical sales role working with customers to help them create IBM solutions to solve their business problems. He holds a BS degree in Math from Towson University in Maryland. He has worked with many IBM products and has extensive experience creating customer solutions using IBM Power, IBM Storage, and IBM System z® throughout his career.

Jordan Antonov is based in Bulgaria and has been working in IBM for the past10 years as part of the IBM Power Systems support team. He is currently part of the Product Engineering team. He has a bachelor 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 solving. He has created multiple technical documents as well as service education offerings.

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

© Copyright IBM Corp. 2025.

Ian Bellinfantie is the Senior IBM AIX and Red Hat Linux SME in the IBM Systems & Software Solutions Group for Saudi Business Machines (SBM), General Marketing & Services Representative of IBM WTC. He has 29 years' experience provisioning IBM Power Systems and before SBM he was working as an IBM contractual employee in the UK. During his career, he has acquired significant experience in the deployment of IBM software & hardware solutions as well as skills in programming and automation in shell, Perl, C++, Java, Chef and Ansible. He holds a degree in Computer Science from the University of Greenwich and a Master's in Financial Markets & Derivatives from the London Metropolitan University. His areas of expertise include the following skillsets: IBM PowerVM, IBM PowerVC, IBM PowerHA, Red Hat Enterprise Linux, Red Hat Ansible, Red Hat OpenShift, AIX and IBM Storage Scale. These skillsets have predominantly been experienced and deployed within the IBM Power systems install base in the Telco, Banking, Retail and Public sectors. Ian brings these skills, experience and insight to this IBM Redbook residency.

Andre Casagrande is a Modernization Specialist with IBM Brazil, focusing on helping customers modernize their IBM Power infrastructure. With over 20 years of experience in the tech industry, Andre is passionate about Linux, AIX, and Free Software projects. He specializes in DevOps, automation tools like Ansible and Terraform, and driving digital transformation by transitioning monolithic applications to microservices. Andre is also responsible for demonstrating products and integrations, including automation tools (Ansible, Terraform) and observability solutions like IBM Instana™. Andre joined IBM in 2022, but his journey with the Power platform started back in 2008 when he worked with a p520 server, sparking his enthusiasm for IBM technologies. Since then, he has contributed to various product implementations, such as OpenShift on IBM Power, PowerVC, IBM PowerVM, VIOS, and HMC. He has also played a key role in Infrastructure as Code (IaC) projects, helping clients automate their infrastructure environments. Andre holds certifications as an RHCA (Red Hat Certified Architect) and SRE (Site Reliability Engineer) through IBM, showcasing his expertise and commitment to advancing the field.

Carlo Castillo is a Client Services Manager for Right Computer Systems (RCS), an IBM Business Partner and Red Hat partner in the Philippines. He has over thirty years of experience in pre-sales and post-sales support, designing full IBM infrastructure solutions, creating pre-sales configurations, performing IBM Power installation, implementation and integration services, and providing post-sales services and technical support for customers, as well as conducting presentations at customer engagements and corporate events. He was the very first IBM-certified AIX Technical Support engineer in the Philippines in 1999. As training coordinator during RCS' tenure as an IBM Authorized Training Provider from 2007 to 2014, he also administered the IBM Power Systems curriculum, and conducted IBM training classes covering AIX, PureSystems, PowerVM, and IBM i. He 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 who is based in Brazil. Previously, he was an employee of IBM, where he worked as a pre-sales technical resource on IBM Power servers. He has 19 years of IT experience, and has worked on various infrastructure projects, including design, implementation, demonstration, installation, and integration of solutions. He has worked with various software on the IBM Power platform, such as PowerVM implementations that include Shared Ethernet Adapter and virtual network interface card, PowerVC, PowerSC, Red Hat OpenShift, Ansible, and Network Installation Manager (NIM) server. During his career at IBM, he served as a consultant for large clients regarding IBM Power and AIX®, performed pre-sales and post-sales activities, and performed presentations and demonstrations for clients. He has worked in several areas of infrastructure during his career and became certified in several of these technologies, such as Cisco CCNA, Nutanix 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, based in the UK. With 28 years of technical and management experience working for IBM Business Partners and customers, he works closely with Offering Management and Development Leaders to deliver successful first-of-a-kind projects and early adoption programs. Paul spearheaded the .NET launch on Power, created the OpenShift Multi-Arch Compute Early Adoption Program, and collaborated with the Development and Research Team's Co-Creation project, delivering the first .NET on Power and OpenShift Multi-Arch Compute Public References. He also regularly presents at conferences, shares knowledge and skills using social media, and has co-authored the “Red Hat OpenShift V4.X and IBM Cloud® Pak on IBM Power Systems Volume 2" Redbook. In addition, he has received two Outstanding Technical Achievement Awards.

Rohit Chauhan is an IBM Champion and a Senior Technical Specialist with expertise in IBM i platform and IBM Power Systems at Tietoevry Tech Services, Stavanger, Norway, which is an IBM Business Partner and one of the biggest IT service providers in the Nordics. He has over 12 years of experience working on the IBM Power platform with design, planning, and implementation of IBM i infrastructure, which includes high availability and disaster recovery (HADR) solutions for many customers during this tenure. Prior to his current position, Rohit worked for clients in Singapore and the UAE in the technical leadership and security role for the IBM Power domain. He possesses rich corporate experience in designing solutions, implementations, and system administration. He is a member of Common Europe Norway with strong focus on the IBM i platform and security. He holds a bachelor’s degree in Information Technology. He is an IBM certified technical expert and also holds an ITIL CDS certificate. His areas of expertise include IBM i, IBM HMC, security enhancements, IBM PowerHA®, systems performance analysis and tuning, Backup Recovery and Media Services (BRMS), external storage, PowerVM®, and solutions to customers for the IBM i platform. Additionally, he is a co-author of several IBM Redbooks publications.

Bartlomiej Grabowski is an IBM Champion and a Principal Systems Specialist in DHL IT Services supporting IBM Power Systems around the world. He has over 20 years of experience in enterprise solutions. He 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 Systems, IBM i, IBM PowerHA, IBM PowerVM, system performance, and storage solutions. Bartlomiej runs technical blog www.theibmi.org. He is a Platinum IBM Redbooks author.

Mithun H R is a Client Technical Architect at IBM Systems development Labs India. He has around 14 years of experience helping clients in modernization and digital transformation. His expertise is on performance optimization, benchmarking, complex integration, Modernization along with Red Hat OpenShift and development. He has experience in Retail, Aerospace, and Defense functional domains. In his current role he is helping Independent Software Vendors adopt cloud platforms and services and develop new workloads and benchmarks to establish competitive advantage of the platforms.

Hemraj Joshi is an accomplished IBM Enterprise Systems Presales Consultant at Gulf Business Machines (KBM) in Kuwait, with over 19 years of experience in IBM technologies. He holds a Master’s Degree in Computer Science from Pune University, India. Specializing in IBM Power Systems, IBM Storage Systems, IBM AIX, IBM PowerVM, IBM PowerHA System Mirror, and Storage Area Networks (SAN). Hemraj has extensive hands-on experience in the design, implementation, demonstration and integration of enterprise solutions. His technical expertise covers a broad spectrum of infrastructure projects, including the implementation of IBM and Brocade Systems and IBM Backup Solutions. In his career, Hemraj has also served as a Subject Matter Expert (SME), contributing his deep knowledge and insights to a wide range of projects. He holds several IBM certifications and an Enterprise Framework Certification such as The Open Group TOGAF, reflecting his commitment to continuous professional development. Currently, Hemraj focuses on technical presales, working closely with clients to craft tailored IBM solutions that address their specific business challenges and goals. His ability to bridge technical expertise with client needs has made him a trusted advisor in delivering impactful solutions.

Niels Liisberg has developed IBM i middleware used in numerous applications worldwide. He is the architect of the IceBreak application server for IBM i. Over the years he has contributed to the IBM i community with presentations, demos, and open-source projects: ILEastic, ILEvator and noxDB to name a few. IT Transformation, modernization, and tools around this process are his passions and he moves anything from 5250 into an era of microservices and containers. He is also a member of the Common Europe Advisory Council (CEAC) and has been an IBM Champion since 2019.

Arnold Ness is a Senior Power Technology Sales Leader in Canada. He has 40 years of experience in IBM and Ciena working with customers designing and implementing business solutions leveraging technology. He holds an MBA in Information Technology Management from Royal Roads University, an Electrical Engineering degree from the University of Alberta and completed IBM's Client Executive program at Harvard. His areas of expertise include solution design and development across x86, IBM Power and IBM 390 platforms. He was awarded the Lou Gerstner Award for Client Excellence in 2022 and has implemented solutions with clients across North and South America. His interests lie in technology innovation, sustainability, Hybrid cloud, Artificial Intelligence, Cybersecurity, Internet of Things (IoT), and Quantum-Safe Computing.

Anil Patil is an Executive Architect and Solutioning Leader in Hybrid Cloud and Data within IBM Consulting™, US. He is a Certified Thought leader in Architect and Solution Consultant community with 25 years of IT experience in design, development, architecture, and Cloud migration for large and complex deal solutions. His core experience is in Generative AI, Red Hat OpenShift, Amazon Web Services (AWS) and Mainframe Modernization. Anil is an IBM Redbooks publication author for different Redbooks and technical contributor for various IBM materials, external publications, and blogs. Anil holds a BE degree in Electronics and Executive MBA in finance and strategy from Rutgers Business School, New Jersey.

Matthew Radford is a remote technical support specialist working in Power HW support from the United Kingdom. He has been in IBM for 28 years working in support. He holds a BSC (Honors) degree in Information Technology from the University of Glamorgan. His area of expertise includes 15 years of supporting AIX and PowerHA, and 2 years working in IBM Power HW support. He has co-authored previous Redbook publications on IBM PowerHA, and is an IBM Redbooks Gold Author.

Sachin Sant is a Linux on Power QA Architect at IBM in India. He has over 26 years of extensive experience in Operating System software development (Linux, IBM AIX, & IBM OS/2), server hardware (IBM Power & Z) validation domain, and open source-based test development and methodologies. He is an active contributor to various open source-based communities, including Linux kernel and Linux test automation projects. In his current role, Sachin is responsible for the functional quality of supported Linux releases on IBM Power servers. Before this, he was the lead for Linux on Z system software test and was also part of the Linux on Power Reliability, Availability, and Serviceability software development team. He holds a degree in Electronics Engineering from Nagpur University, India.

Mick Tarsel is a Senior Linux Cloud Engineer with a heavy focus on Linux networking. He has gained over 10 years of experience with virtualization software on IBM Power Systems with extensive skills configuring, debugging, and optimizing network connectivity for various types of data centers. He is an active contributor to open-source development. In 2019 Mick was published in the ICES Journal of Marine Science for his work related to using open-source software to better conserve coral reefs around the world. He enjoys volunteering at high school's teaching the importance of open-source collaboration and GNU/Linux principles.

Additionally, he enjoys enabling college students to build and test open-source software on Power Systems. His interests lie in software defined networking, operating system development, network security, and virtualization technologies.

Andy Youens is the Managing Director of UK based FormaServe Systems, an IBM ISV since 1990. Andy is a seasoned IBM i professional with over 40 years of experience in the field. He has a deep expertise in IBM i systems, including system administration, application development and modernization. Andy's extensive knowledge and hands-on experience have made him a sought-after expert in the IBM i community, both at home and internationally. As an active contributor to the IBM i community, Andy frequently shares his insights and expertise through speaking engagements, webinars and technical articles. An IBM champion, who co-wrote the IBM i developer certification. He is the owner of PowerWire, the monthly newsletter and articles on IBM Power. Prior to his IT career, Andy is proud to have served in the UK's armed forces, the Royal Navy.

Henry Vo is an IBM Redbooks Project Leader with 11 years of experience in IBM. He has technical expertise in business problem solving, risk/root-cause analyze, and writing technical plans for business. He has had multiple role in IBM such as Project management, ST/FT/ETE Test, Back End Developer, DOL agent for NY and is certified in IBM zOS Mainframe Practices, IBM Z®¬¨ÐÜ System programming, Agile, and Telecommunication Development Jumpstart. Henry holds a Master of MIS (Management Information System) from the University of Texas at Dallas since 2012.

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

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

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

Ishwar Fernandes
IBM Champion and CSI 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, TX

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

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, renowned for their exceptional reliability and performance, have long been the backbone for critical business applications running on AIX, IBM i, and Linux. As organizations accelerate their digital transformation, application modernization becomes essential for enhanced agility, cost reduction, and innovation. IBM Power provides a robust platform for this journey, enabling the transformation of legacy systems while preserving its inherent reliability. With strong support for modern containerization, including Red Hat OpenShift, and a powerful processor architecture capable of processing significantly more workload per core than competitors, IBM Power is ideally suited for modern environments.

This section of our Redbook explores how IBM Power empowers clients in their modernization efforts, showcasing client examples, use cases, and the tools available for managing modernized application stacks.

The following chapters are included in this part:

© Copyright IBM Corp. 2025.

1

Defining Modernization

Modernization refers to the process of upgrading and refining existing systems and applications to boost performance, efficiency, security, and overall value. As businesses increasingly rely on technology to connect with customers, streamline processes, and stay competitive, meeting customer expectations for seamless digital experiences becomes critical. Legacy systems, however, can impede this, making modernization essential – not just for technological upgrades, but for transforming business operations and aligning IT infrastructure with evolving business needs.

As organizations generate large volumes of data, modern systems become essential for processing, analyzing, and utilizing this information effectively. The rise of cloud platforms offers businesses scalability, flexibility, and cost efficiency that traditional on-premises systems struggle to match. Additionally, the changing security landscape demands stronger protection, often absent in outdated systems.

By modernizing applications and infrastructure, businesses can achieve enhanced agility and innovation, enabling faster time-to-market for new products and services. Modernization also enhances the customer experience, offering more personalized, responsive, and engaging interactions. It reduces costs by improving resource use and minimizing downtime, while strengthening security to better protect against cyber threats.

With a well-planned modernization strategy, organizations can unlock significant value from their tech investments, thriving in the digital age and gaining a competitive advantage by leveraging the latest technologies.

The following topics are included in this chapter:

© Copyright IBM Corp. 2025.

1.1 Introducing modernization

In today’s fast-paced digital world, businesses are under constant pressure to innovate, improve efficiency, and stay ahead of the competition. One of the most effective ways to achieve these goals is through IT modernization – the process of upgrading, transforming, and optimizing an organization’s technology infrastructure and applications. IT modernization goes beyond simply adopting the latest technologies; it involves rethinking how businesses operate, aligning IT systems with evolving business needs, and ensuring seamless, secure, and scalable digital experiences.

As organizations increasingly rely on technology to drive operations, manage data, engage customers, and deliver services, the limitations of legacy systems become more apparent. Outdated systems can hinder performance, compromise security, and lead to inefficiencies, making it harder for companies to respond quickly to market changes. IT modernization helps bridge this gap by improving agility, reducing costs, enhancing security, and enabling businesses to deliver better customer experiences.

Whether through cloud adoption, application updates, or the integration of emerging technologies, IT modernization is essential for any organization looking to thrive in the digital age. By embracing modernization, businesses can unlock new opportunities, enhance operational capabilities, and position themselves for long-term success.

1.2 What is driving IBM clients to modernize

What are the drivers for IBM clients to modernize their IBM Power infrastructure?

Figure 1-1 provides picture into what clients are looking for in their modenrized IT environment.

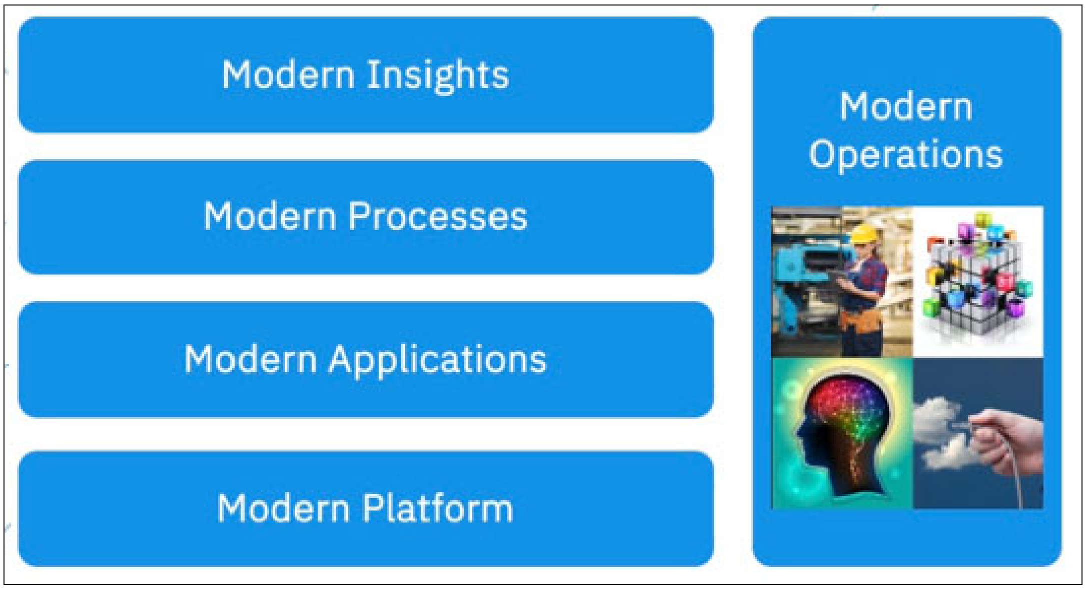


Figure 1-1 Drivers for modernization

Modern insights

In today's fast-paced market, our clients are driving modernization to unlock the power of their data. They seek a unified data platform that supports seamless sharing, real-time analytics, and deeper organizational insights. This enhanced data access allows for proactive decision-making, strengthening client and supplier relationships and ensuring they stay ahead of the competition.

Modern processes

In today's rapidly evolving landscape, IBM clients are challenged to maintain pace. They require automation driven by AI and rule-based systems to drastically reduce response times, minimize decision-making risks, and empower them to adapt swiftly to market changes.

Modern applications

Struggling with cumbersome monolithic applications, IBM clients are seeking to enhance user experiences and drastically reduce time-to-market. They require a shift to distributed, API-enabled functions, allowing for faster development cycles, easier integration of new technologies, and ultimately, more adaptable and user-centric applications.

Modern platform

IBM clients are prioritizing infrastructure transformation to achieve significant TCO reduction and embrace flexible consumption models. They seek to minimize CapEx investments, build a sustainable infrastructure footprint, and bolster security against escalating cyber threats. Furthermore, they aim to integrate advanced AI technologies to enhance application performance and streamline business processes.

Modern operations

To remain competitive in a rapidly changing world, clients must modernize operations. This enables them to swiftly react to legislative and social shifts, adapt to evolving user requirements, and reduce IT infrastructure costs. Modern application development is the key to achieving faster time-to-market, improved management, and reduced risk.

1.3 Benefits of modernization

Modernization allows organizations to improve their competitiveness by increasing agility, lowering costs, and utilizing existing IT investments. The priority for modernization has increased, with 75% of organizations identifying it as a critical initiative.[1] Studies show that organizations implementing modernization strategies achieve a 64% increase in code releases and a 44% reduction in time per release.[2]

Cloud migration strategies vary. Lift and shift migrations often fail to meet cloud adoption goals.[3] A structured approach helps organizations determine whether rehosting, replatforming, or refactoring is the most effective path. Applications using distributed computing patterns and service-oriented architectures require modifications[4], improving cloud transition speed and cost efficiency.

Modernization also reduces costs. Organizations that optimize their IT environments achieve a 19% reduction in operational costs over five years.[5] Additionally, modernized infrastructures reduce unplanned outages, improving availability and mitigating reputational risk. Addressing architectural challenges, such as dynamic provisioning and latency, further enhances performance.

[1] Cloud Pulse Survey, September 2019

[2] Application Modernization Services Market – Global Forecast to 2025

[3] Gartner, Use Cloud-Native Architecture to Modernize Your Applications, July 2024

[4] IBM, IBM Cloud Paks, 2024.

[5] Forbes, IBM Acquires Advanced's Application Modernization Division, January 2024.

Scalability, resilience, and security are key modernization outcomes. Cloud-native strategies accelerate application delivery, reduce time to market, and increase business agility. These improvements enable organizations to adapt faster to customer demands and regulatory changes.

The following sections discuss benefits of modernization from a business, technical, operational, and data-driven perspective.

1.3.1 Business benefits

Modernization promotes business growth by enabling organizations to promptly adapt to market changes and address evolving customer expectations. Figure 1-2 details the principal business benefits.

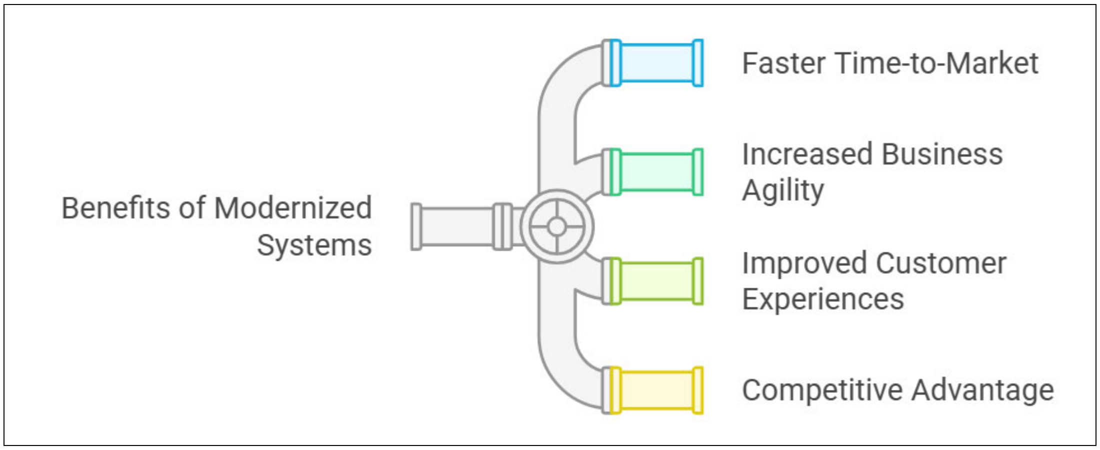

Benefits of Modernized Systems
Faster Time-to-Market
Increased Business Agility
Improved Customer Experiences
Competitive Advantage

Figure 1-2 Benefits of modernized systems

Faster time-to-market

Enable quick development and deployment. Using CI/CD pipelines and automation, organizations can rapidly meet market demands while maintaining high quality.

Increased business agility

Adapt to changing needs, allowing organizations to scale, pivot strategies, and integrate new technologies, thus responding to market shifts, customer demands, and regulatory changes.

Improved customer experiences

Improves application reliability, performance, and usability, enabling faster responses, intuitive interfaces, and personalized services, leading to higher satisfaction and loyalty.

Competitive advantage

Organizations can achieve rapid innovation, industry differentiation, and market leadership by integrating modern technologies such as artificial intelligence, hybrid cloud solutions, and advanced analytics.

1.3.2 Technical benefits

Performance is enhanced, interoperability is improved, and resilience is ensured through modernization. These benefits allow organizations to integrate new technologies and optimize existing infrastructure.

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

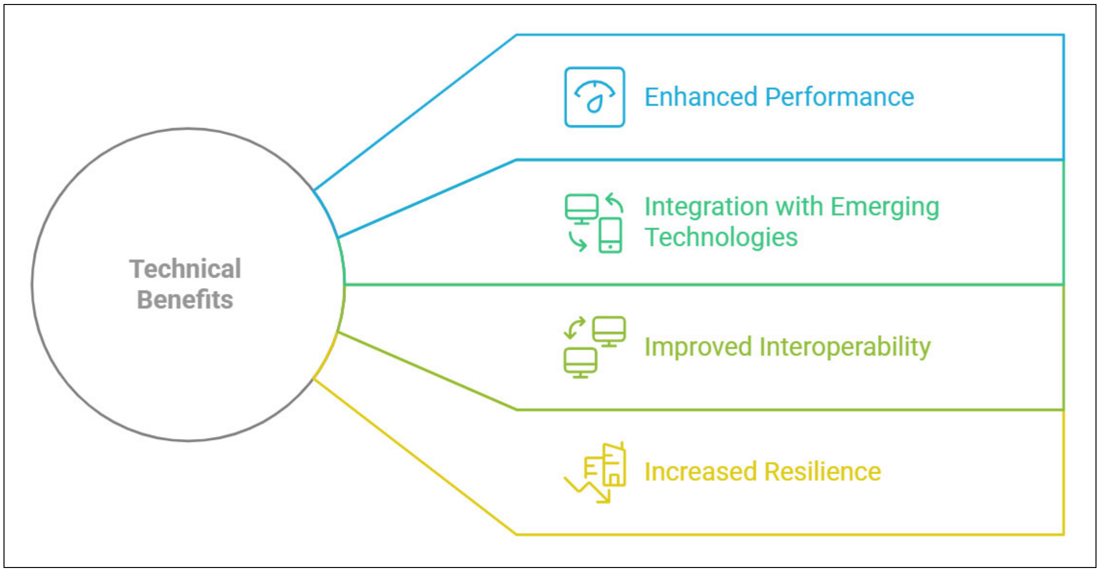

Technical Benefits
Enhanced Performance
Integration with Emerging Technologies
Improved Interoperability
Increased Resilience

Figure 1-3 Technical benefits of modernized systems

Enhanced performance

Modern architecture allows systems to maximize resource utilization. By adopting modern frameworks, organizations can achieve faster processing speeds, improved reliability, and better scalability for applications and workloads.

Integration with emerging technologies

Integration with advanced technologies, such as artificial intelligence, machine learning, and hybrid cloud services, is supported by modernized environments. This integration enables organizations to utilize cutting edge tools to improve operations and innovate.

Improved interoperability

Improved interoperability ensures that various systems and tools can communicate effectively, thereby simplifying processes and minimizing operational silos. Modern systems are created to function across different platforms, applications, and services.

Increased resilience

To handle disaster recovery, maintain high availability, and provide fault tolerance, today's infrastructure is designed to enable continuous operations during failures or disruptions.

1.3.3 Cost optimization

Modernization delivers cost optimization benefits, enabling organizations to reduce operational expenses while improving efficiency and sustainability.

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

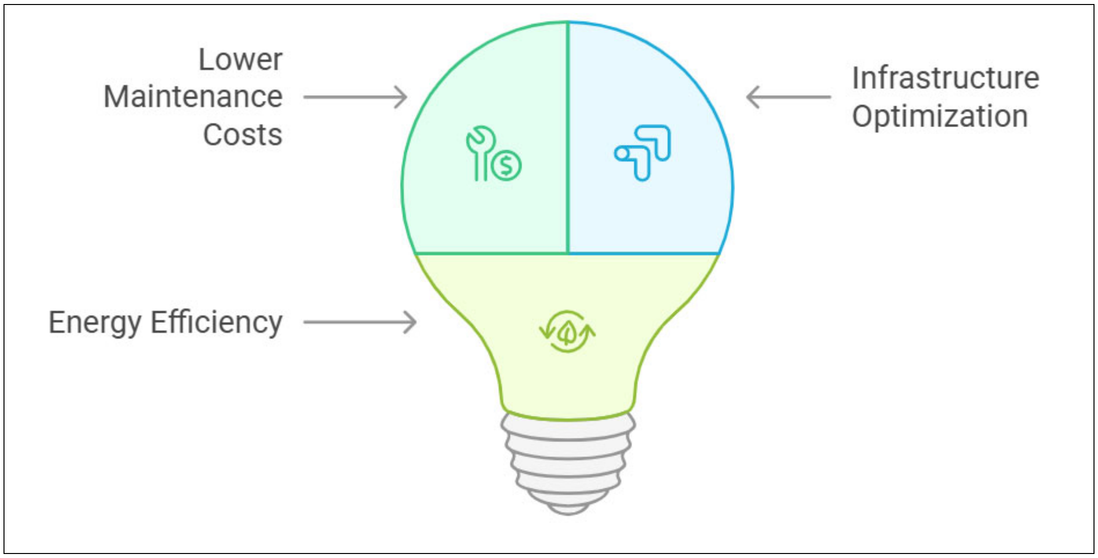


Figure 1-4 Cost Optimization benefits of modernized systems

Lower maintenance costs

Reducing reliance on older hardware and software, which can be costly to maintain, transitioning to modern infrastructure allows organizations to minimize maintenance expenses. This shift also optimizes capital expenditures (CAPEX), enabling more allocation of resources towards other strategic goals.

Infrastructure optimization

Improved utilization of computing, storage, and networking resources is facilitated by recent advancements. Technologies such as virtualization and containerization enable organizations to optimize their infrastructure for scalability and flexibility, thus reducing the Total cost of ownership (TCO).

Energy efficiency

Modern infrastructure uses less energy and delivers better performance. Upgrading to efficient hardware and software reduces energy use, cuts utility costs, and supports sustainability goals.

1.3.4 Operational benefits

Operations, security, and management are all enhanced by modernized systems, which improve overall performance and control.

Figure 1-5 shows the operational benefits of modernization.

Centralized Management
Workflow Optimization
Improved Security

Figure 1-5 Operational benefits of modernized systems

Centralized management

Modernization allows IT teams to manage multiple environments from one control plane. Using tools such as IBM Cloud Paks and automated monitoring, organizations can simplify administration, lower management costs, and increase system visibility.

Workflow optimization

Automation and optimized processes are employed to enhance workflow. 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, and continuous monitoring enhance security. Organizations acquire robust frameworks that comply with industry standards, protecting data and systems from emerging threats. Examples include using encryption standards like AES-256 and implementing multi-factor authentication protocols.

1.3.5 Data-driven decision making

Organizations benefit from tools that provide insights and facilitate data integration across platforms. Figure 1-6 illustrates key advantages: maximizing data value and enhancing data portability.

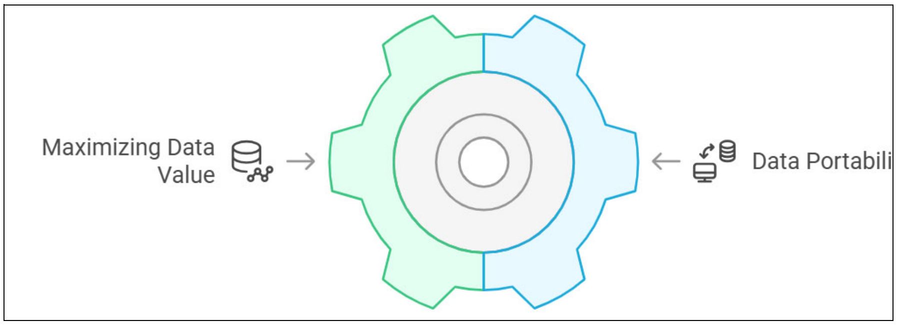


Figure 1-6 Data-driven benefits of modernized systems

Maximizing data value

Modernized environments enhance enterprise data utility by using advanced analytics and AI. This allows organizations to gain insights from large datasets, make data-driven decisions, optimize operations, and identify business opportunities. IBM Watson® Analytics®, for instance, can forecast market trends or detect inefficiencies with its predictive analytics capabilities.

Data portability

Facilitates data transfer and integration across a variety of 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. This capability also aids in meeting compliance requirements by data access and reporting across different regions and systems.

1.4 Modernization in IBM Power

IBM Power is a family of systems that are capable of running mission-critical workloads utilizing hybrid multicloud technologies. IBM Power servers are high-performance, secure, and reliable servers built on the IBM Power processor architecture. Figure 1-7 shows the Power10 family starting with the Power10 one and two socket scale-out servers, scaling to the four socket Power E1050 and then to the eight socket Power E1080.

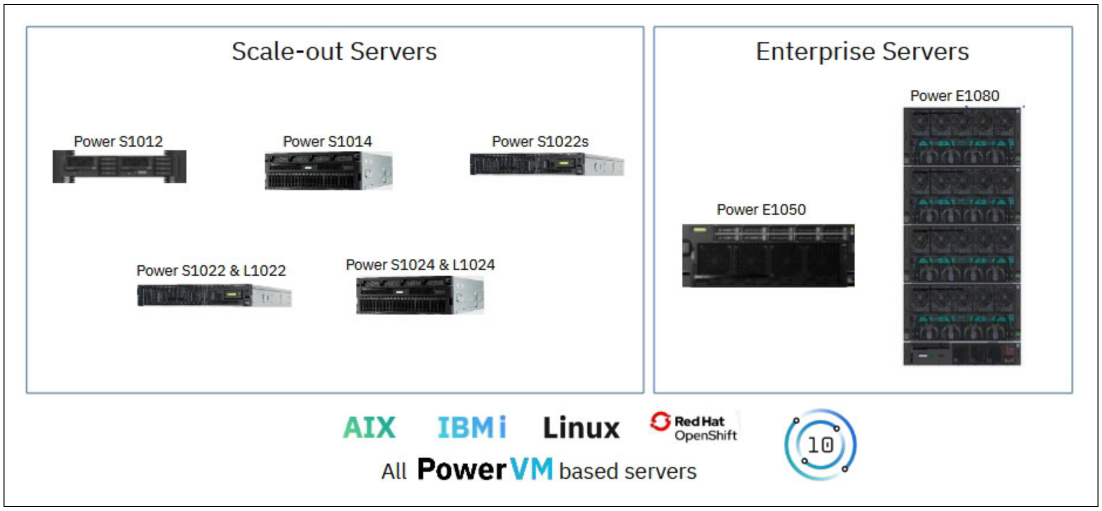


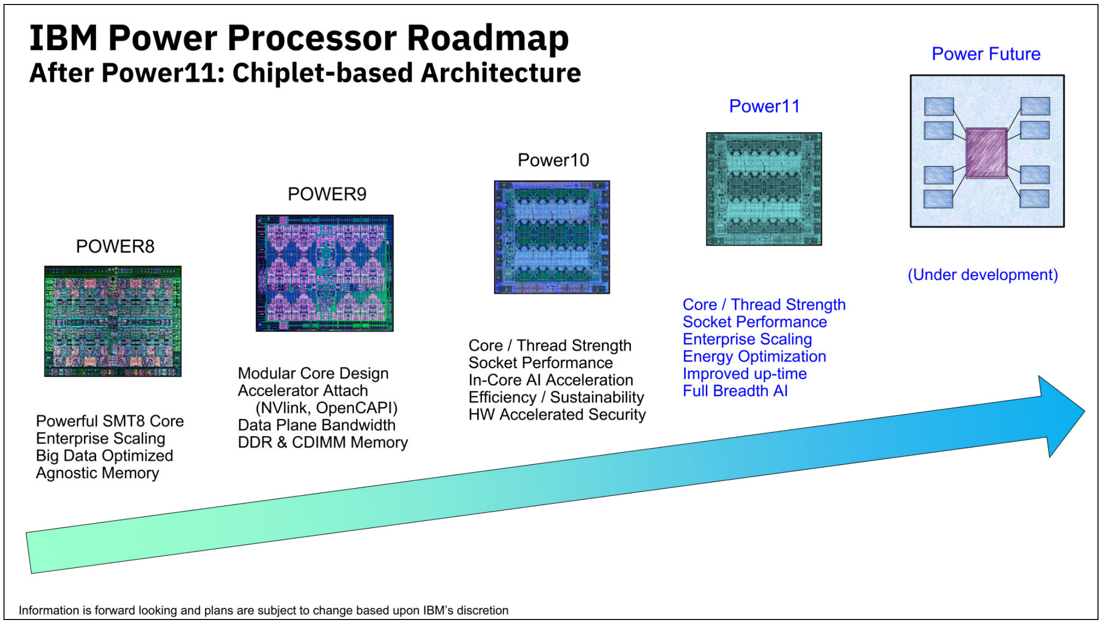
Figure 1-7 The Power10 family

IBM's Power processor roadmap showcases a 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 those shown in Figure 1-7, are designed to handle critical workloads with improved core-to-cloud data protection and streamlined automation These servers offer up to 75% performance improvements for the same workloads compared to previous generations[6]. Key features include multiple packaging options including the Single Chip Module (SCM used in the E1080), Dual Chip Module (DCM used in the midrange and scale models), and Entry Single Chip Module (eSCM used in the low end scale out models) packaging, PCIe Gen5 connectivity, and transparent memory encryption The Power10

[6] https://www.ibm.com/power

architecture also supports advanced AI capabilities through the Matrix-Math Assist (MMA) feature, which enhances matrix multiplication operations for use in AI.

Looking ahead, the Power11 processor, set to be released in 2025, promises even greater advancements. IBM's Spyre™ Accelerator, designed for efficient AI computation, will be integrated into Power11 systems, significantly boosting AI processing capabilities. The Power11 architecture builds upon the reliability, availability, and serviceability (RAS) characteristics of Power10, while introducing improved energy efficiency and quantum-safe security With these enhancements, Power11 servers are poised to support emerging enterprise AI use cases and propel digital transformation initiatives for mission-critical infrastructure. Figure 1-8 shows the recent generations of the IBM Power processor based systems and shows that IBM is continuing to invest in future generations.



IBM Power Processor Roadmap

After Power11: Chiplet-based Architecture

POWER8

Powerful SMT8 Core
Enterprise Scaling
Big Data Optimized
Agnostic Memory

POWER9

Modular Core Design
Accelerator Attach
(NVlink, OpenCAPI)
Data Plane Bandwidth
DDR & CDIMM Memory

Power10

Core / Thread Strength
Socket Performance
In-Core AI Acceleration
Efficiency / Sustainability
HW Accelerated Security

Power11

Core / Thread Strength
Socket Performance
Enterprise Scaling
Energy Optimization
Improved up-time
Full Breadth AI

Power Future

(Under development)

Information is forward looking and plans are subject to change based upon IBM's discretion

Figure 1-8 IBM Power family history and roadmap

Many of the world's most mission-critical enterprise workloads are run on IBM Power. The core of the global IT infrastructure, encompassing the financial, retail, government, health care, and every other sector in between, is comprised of IBM Power systems, which are renowned for their industry-leading security, reliability, and performance attributes. For enterprise applications, including AI, ERP, databases, application and web servers, many clients use IBM Power.

Enterprise IT delivery is changing as a result of digital transformation – cloud computing is playing a key role. When it comes to consuming infrastructure, you need options and flexibility, and IBM Power is designed to run in your datacenter utilizing flexible cloud capabilities or in the cloud. Whether you are using Red Hat OpenShift and Kubernetes to modernize enterprise applications, creating a private cloud environment within your data center using adaptable pay-as-you-go services, using IBM Cloud to launch applications as needed, or creating a seamless hybrid management experience across your multicloud landscape – IBM Power is fully capable of delivering whatever hybrid multicloud approach you choose.

The modern data center consists of a combination of on-premise and off-premise, multiple platforms, such as IBM Power, IBM Z and LinuxOne, and x86. Applications range from monolithic to cloud-native – which is inherently some combination of bare metal, virtual machines, and containers. An effective hybrid cloud management solution must account for all of these factors. IBM and Red Hat are uniquely positioned to best accommodate the applications that you are running today and the modernized applications of tomorrow, wherever they reside.

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

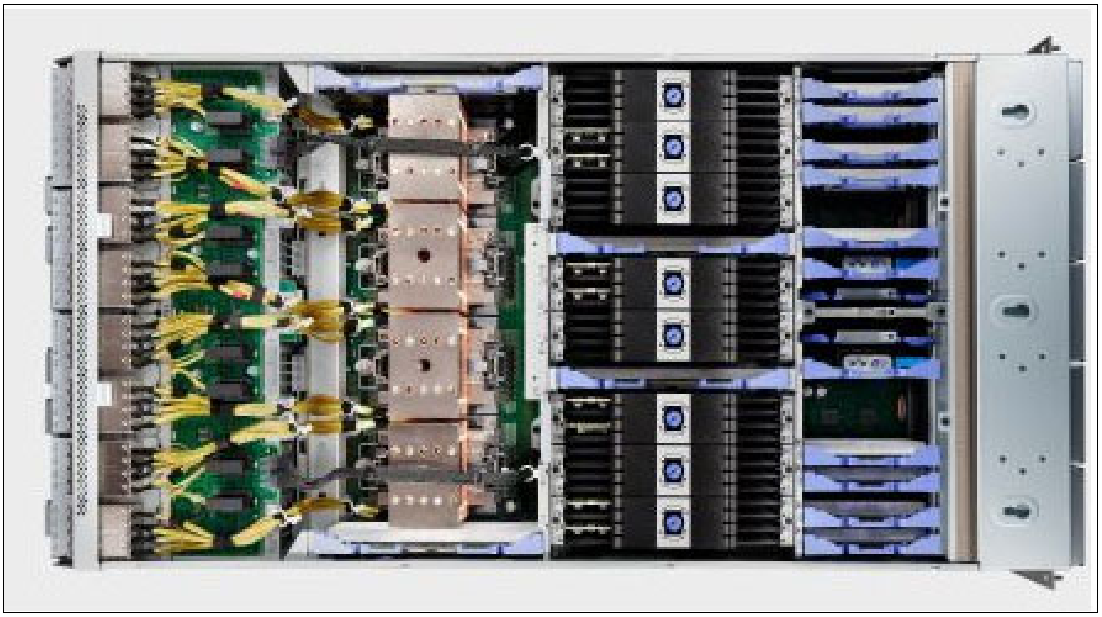


Figure 1-9 View of Power E1080 central electronic component drawer

IBM Power delivers one of the highest availability ratings among servers

According to the ITIC 2023 Global Server Hardware, Server OS Reliability survey[7], which polled nearly 1,900 corporations worldwide across over 30 vertical market segments, an 88% majority of the newest IBM Power10 server users (shipping since September 2021) say their organization achieved eight nines--99.999999% of uptime. This is 315 milliseconds of unplanned outage time, per server, per year due to underlying system flaws or component failures (second only to IBM Z with 31.56 milliseconds of per server annual downtime). So, Power10 corporate enterprises spend just $7.18 per server/per year performing remediation due to unplanned server outages that occurred due to inherent flaws in the server hardware or component parts.

This marks 2023 as the 15th consecutive year that the IBM Z/LinuxOne and IBM Power Systems have dominated with the best across-the-board uptime reliability ratings among 18 mainstream distributions.

This level of availability is largely due to the inherent availability features of Power10, allowing for less downtime than comparable offerings, due to built-in recovery and self-healing functions for redundant components. Organizations are also able to switch from an earlier Power server to the current generation while applications continue to run, giving you high-availability and minimal downtime when migrating.

[7] https://itic-corp.com/itic-2023-reliability-survey-ibm-z-results/

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

For the fourth straight year, IBM Power has been rated as one of the most secure systems in 2022[8], with 2.7 minutes or less of unplanned outages due to security issues. This means that IBM Power is:

- 2x more secure than comparable HPE Superdome servers,
- 6x more secure than Cisco UCS servers,
- 16x more secure than Dell PowerEdge servers,
- 20x more secure than Oracle x86 servers
- up to more than 60x compared to unbranded white box servers.

Security breaches were also detected immediately or within the first 10 minutes in 95% of the IBM Power systems that were surveyed. This results in better chances that a business will suffer little to no downtime, nor will they be susceptible to damaged, compromised, or stolen data.

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

A recent user case study illustrated how IBM Power enabled a customer to increase end-user application performance by 20%, ending up with them meeting their sustainability goals. By helping move the customer to IBM Power and IBM FlashSystem® storage, they were fully able to leverage their SAP S/4HANA operations enabling them to meet their climate objectives.

IBM Power streamlines AI operations with advanced on-chip technologies

IBM Power systems delivers 5X faster AI inferencing per socket for high precision math over the previous generation. This is accomplished through multiple Matrix Math Accelerator (MMA) units in each Power processor core. MMAs allow IBM Power systems to forgo external accelerators, such as GPUs and related device management, when running machine learning and inferencing workloads.

Current IBM Power systems can range from scale-out servers that start with 1 core and 32 GB of memory on the IBM Power S1012 to enterprise systems with up to 240 cores and 64 TB of memory on the IBM Power E1080.

> **Note:** The full lineup of IBM server models based on the latest Power processors can be found at IBM Power.

1.4.1 Operating system support in IBM Power

IBM Power offers unparalleled flexibility by enabling you to consolidate diverse operating environments onto a single system. From industry-leading options like AIX and IBM i, to the widely adopted Linux and Red Hat OpenShift platforms, you can harness the power of IBM Power to consolidate mission-critical applications across any number of systems.

This provides:

- Enhanced reliability, availability, and security.
- Ability to respond faster to business demands.

[8] 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.

- Protection for your data from core to cloud.
- Streamlined insights and automation.

Modernize your applications and infrastructure with a frictionless hybrid cloud experience. IBM Power servers provide the agility, reliability and sustainability your organization requires.

Supported Operating Systems

As of the time of this publication, Power10 processor-based systems support the following platforms/operating system versions shown 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	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 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	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 used in the table above is the Power E1080. Software maps detailing which versions are supported on which specific IBM Power server models (including previous generations of IBM Power) can be found in its own IBM Support page.

A full list of supported operating systems can also be found here.

AIX

IBM AIX is IBM’s proprietary Unix operating system designed to run on IBM Power servers. The very first Power processors on what was then known as the RISC System/6000 (RS/6000) ran AIX v3, although earlier versions of AIX ran on earlier IBM hardware (e.g., the RT/PC). The currently available versions of AIX on IBM Power as of this writing are AIX v7.2 and v7.3.

For over three decades, organizations have trusted IBM AIX to run their most mission-critical applications. AIX on Power drives innovation with hybrid-cloud and open-source capabilities that help you build and deploy modern applications within a secure and resilient environment.

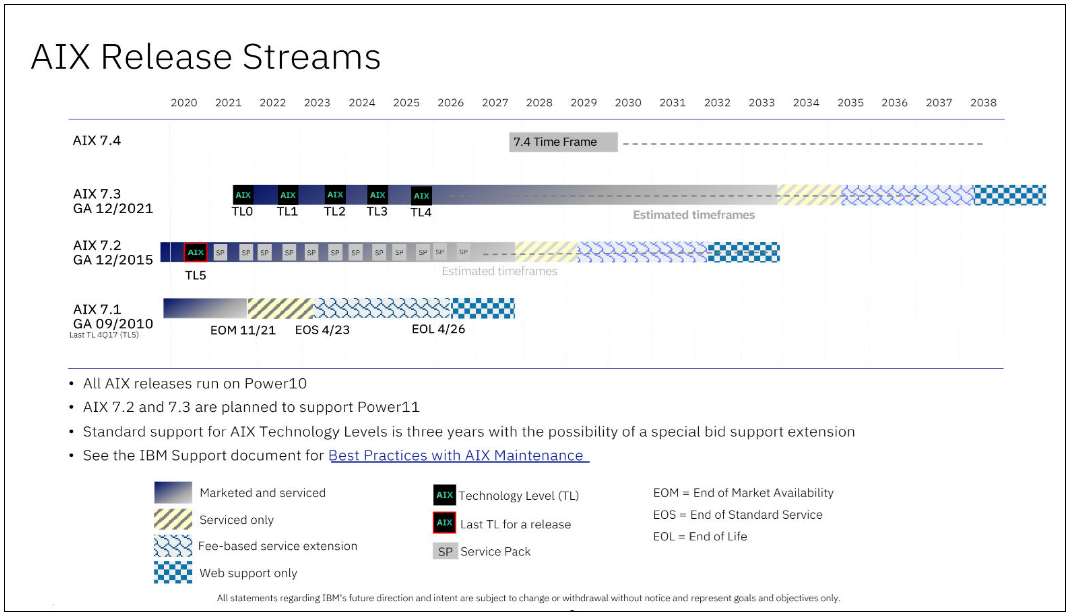
Almost every business today recognizes that digital transformation is critical to better serving its customers, reducing costs and improving operational efficiency, while also stepping up environmental sustainability. To that end businesses are embarking on both infrastructure and application modernization to help drive these improvements – developing or moving more applications to the cloud and adopting new technologies such as containerization. The most successful of these companies also recognize that this journey requires a hybrid cloud approach.

Over the past two years, IBM Power has introduced new offerings to help businesses accelerate this transformation – introducing IBM Power Hybrid Cloud capability featuring on-premises Power Private Cloud with consumption pricing and Power Virtual Server. IBM has also continued to expand its focus on open source technologies, bringing popular open source tools to IBM AIX and creating a collection of Ansible automation packages and playbooks to make managing AIX easier than ever, all with consistent skills and processes on x86-based platforms.

AIX and Power have been the foundation of mission-critical workloads and databases for tens of thousands of customers over the past 35 years, leading the industry in performance, scalability, resiliency, flexibility and security. IBM aims to sustain that platform leadership and continue to evolve and extend AIX to help customers capitalize on new capabilities, including running Red Hat OpenShift containers adjacent to AIX in order to reduce latency or embedding AI inference capability in enterprise applications on AIX.

AIX release roadmap

AIX on Power drives innovation with hybrid-cloud and open-source capabilities that help you build and deploy modern applications within a secure and resilient environment. AIX will continue to be a strategic, foundational component of the portfolio with a roadmap and support plan that extends beyond 2035. A depiction of that roadmap can be seen in Figure 1-10.



- All AIX releases run on Power10
- AIX 7.2 and 7.3 are planned to support Power11
- Standard support for AIX Technology Levels is three years with the possibility of a special bid support extension
- See the IBM Support document for Best Practices with AIX Maintenance

▨ Marketed and serviced
▨ Serviced only
▨ Fee-based service extension
▨ Web support only

AIX Technology Level (TL)
AIX Last TL for a release
SP Service Pack

EOM = End of Market Availability
EOS = End of Standard Service
EOL = End of Life

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

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 provides strong, enduring security with features that include Trusted AIX and Trusted Execution.
- **Unmatched uptime** - Power systems running AIX 7.3 have the lowest percentage of unplanned annual server downtime and best-in-class reliability.
- **Investment protection** - The binary compatibility of AIX OS allows applications to run unchanged and without recompiling on the newest releases, guaranteed.

- **Enterprise AI** - Streamline insights by running AI inference directly in the core with AIX 7.3.

AIX binary compatibility

AIX binary compatibility allows applications that were created on earlier releases or technology levels of AIX to run unchanged and without recompiling on later releases or technology levels of AIX. For example, an application that is created on AIX Version 6.1 can be run on AIX Version 7.2, or later.

The ability to run applications that were created on an earlier version of an operating system on a later level of the operating system is known as compatibility with an earlier version. Applications must use only portable programming techniques for binary compatibility on any platform.

AIX binary compatibility is fully explained in IBM Documentation found at this link: `https://www.ibm.com/docs/en/aix/7.3?topic=aix-binary-compatibility`.

A system that uses AIX 7.3, or later, might operate as a server for client machines that are running an earlier version of AIX. In this case, the server operates only if the necessary compatibility options are installed. All conditions about binary compatibility apply in this scenario.

> **Note:** If applications are not running correctly after you migrate to a newer version of the AIX operating system, you can open a case with IBM support. When you open a case, you must specify AIX Binary Compatibility in the Title field.
>
> There are also restrictions to AIX binary compatibility outlined in this IBM Documentation.

More information on AIX can be found at `https://www.ibm.com/products/aix`.

IBM i

The IBM i operating system, formerly known as AS400 and iSeries, offers a range of benefits that make it a strong choice for businesses, particularly those with mission-critical applications. The lineage of IBM i goes back to some of the earliest IBM midrange systems used for small to medium sized businesses, but its capabilities have grown exponentially. Some of the largest companies in the world use IBM i running on the IBM Power server as their strategic platform for manufacturing planning, retail, distribution, logistics, banking, health care, insurance, hospitality management, government management, and legal case management.

IBM i is a powerful and versatile platform that offers businesses a secure, scalable, and efficient foundation for running their applications and managing their data. Its focus on solutions, openness, and integrated value makes it an attractive choice for organizations looking to leverage the benefits of a robust and reliable operating system and database system.

IBM i is a fully integrated operating system, meaning the database, middleware, security, runtime, and hypervisor are all integrated into the stack and licensed as one. This integration helps clients lower their TCO, simplify systems management, and do more with fewer resources.

Forrester Consulting found that clients deploying IBM i on-premise or in the cloud realized, on average, a 191% return on investment (ROI) and a payback of just 6 months for their business. These clients also saw savings of USD 1.06 million in reduced system downtime and an increase in productivity of USD 470,000 over three years[9].

IBM i Roadmap

A unique feature of IBM i is its backward compatibility. Organizations can run legacy AS/400 applications on IBM i without compatibility issues. This eliminates the need for costly and time-consuming code migrations. IBM i has a strong ecosystem of users and developers with a wide variety of independent solution providers.

Figure 1-11 shows the currently supported versions of IBM i and the roadmap for support for future releases.

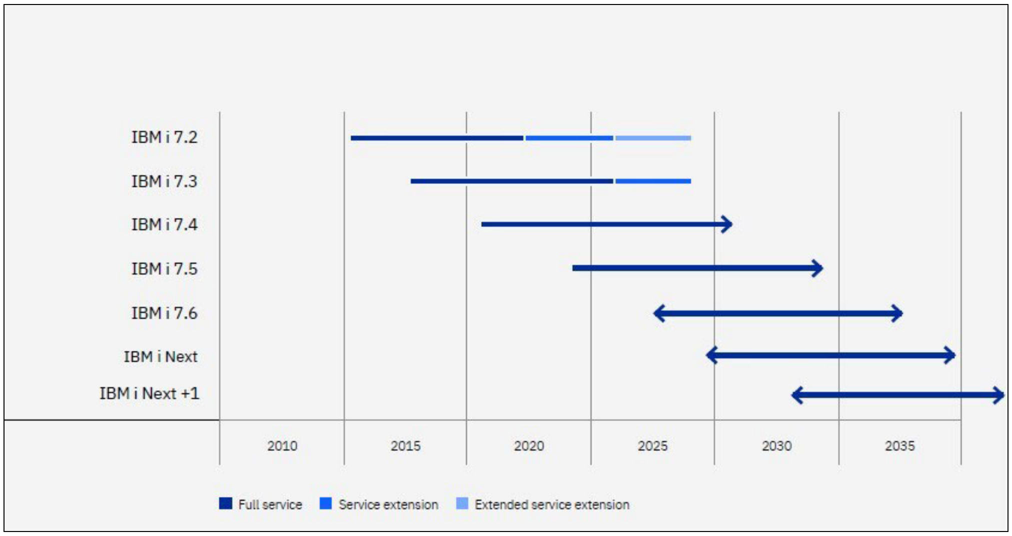


Figure 1-11 IBM i release schedule

Subscription licensing

Subscription-term licensing offers clients easier annual budget planning with predictable and consistent payment options. Software licenses and support are integrated into one subscription price. Annual subscriptions are automatically updated to the latest release, reducing business risks and security vulnerabilities by keeping technology current. With auto-renewal, clients no longer have to worry about losing support or managing new software keys. Clients on older releases can get back on support and current with technology more easily and at a lower price point.

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's a strong choice for businesses looking for a stable and secure platform for their mission-critical applications, while also allowing them to embrace new technologies and adapt to evolving business needs. To explore more on the IBM i operating system please refer to the IBM i product page.

Linux on Power

With IBM systems and Linux open software, you can choose the best hardware for your workloads while controlling software costs. Linux is a highly-rated operating system providing flexibility, stability, and low total cost of ownership. Because Linux is an open source development project, it is constantly improved by community innovation. IBM supports Linux as a long-term strategic platform. IBM participates in the Linux community through the Linux Technology Center (LTC).

[9] Forrester Consulting: The Total Economic Impact Of IBM i
`https://www.ibm.com/account/reg/us-en/signup?formid=urx-52179Forrest`

Linux is certified on all IBM systems – including IBM Power, Z, and LinuxONE – so that you can choose the hardware that makes the most sense for your business. Table 1-2 shows the Linux distributions supported on IBM Power10 systems.

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

<table>
<tr><th>IBM Power10 Systems</th><th>Distributions supported</th></tr>
<tr><td>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)</td><td>Red Hat Enterprise Linux 9.0, any subsequent RHEL 9.x releases
Red Hat Enterprise Linux 8.4, any subsequent RHEL 8.x releases
SUSE Linux Enterprise Server 15 SP3, any subsequent SLES 15 updates
Red Hat OpenShift Container Platform 4.9, or later
Ubuntu 22.04, or later[a]</td></tr>
<tr><td>9080-HEX (IBM Power E1080)</td><td>Red Hat Enterprise Linux 9.0, any subsequent RHEL 9.x releases
Red Hat Enterprise Linux 8.4, any subsequent RHEL 8.x releases
Red Hat Enterprise Linux 8.2 (POWER9 Compatibility mode only)[b]
SUSE Linux Enterprise Server 15 SP3, any subsequent SLES 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]</td></tr>
<tr><td>9028-21B (IBM Power S1012)</td><td>Red Hat Enterprise Linux 9.2, for PowerLE, or later
Red Hat OpenShift Container Platform 4.15, or later
SUSE Linux Enterprise Server 15 SP6, any subsequent SLES 15 updates
Ubuntu 22.04, or later[a]</td></tr>
<tr><td colspan="2">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.
▸ Red Hat Enterprise Linux 8.6, or later: up to 64 TB of memory and 240 processor cores.
▸ Red Hat Enterprise Linux 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.</td></tr>
</table>

[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 on Linux distributions supported on power, please visit this IBM documentation article.

Python packages in Linux on IBM Power

You can minimize the entry barrier for AI by natively using Python packages on Linux on Power LPARS. No container platform needed, allowing for minimal extensions of IBM Db2, SAP, and ORACLE landscapes. Benefit from over 200 packages optimized for IBM Power10.

Note: See these notes on Installing and using RocketCE in a Linux on Power LPAR

SAP RISE

RISE with SAP Methodology provides SAP customers with confidence for a clear journey to SAP Business Suite.[10]

In February 2023, SAP and Red Hat, who is the world's leading provider of open source solutions, announced an expanded partnership to significantly increase SAP’s use of and support for Red Hat Enterprise Linux. This collaboration aims to enhance intelligent business

[10] https://www.sap.com/mena/products/erp/rise.html

operations, support cloud transformation across industries and drive holistic IT innovation.[11] SUSE Enterprise Linux continues to be supported as well.

Red Hat Enterprise Linux (RHEL) is a hardened, production-ready Linux operating system for hybrid cloud innovation, and is trusted by global enterprises across industries worldwide. The platform builds on this trust by offering a consistent, reliable foundation for SAP software deployments, providing a standard Linux backbone to support SAP customers across hybrid and multi-cloud environments. Additionally, SAP will be running a continuously increasing part of its internal IT infrastructure and the SAP Enterprise Cloud Services portfolio on RHEL.

IBM Power servers are purpose built for data-intensive applications such as SAP HANA and S/4HANA that require large amounts of in-memory computing but still let you maintain the high availability and flexibility required for your hybrid cloud.

With global data volumes set to grow to more than 180 zeta bytes in 2025, organizations across every sector are facing tremendous pressure to manage, process, store and extract valuable insights from their critical data. By running SAP HANA on IBM Power servers, businesses can reduce datacenter costs and enhance environmental sustainability.

- The IBM Power E1050 requires half the amount of energy (50% less energy) used by compared x86-based servers of comparable performance.
- The IBM Power E1080 uses 15% less energy and provides 54% more performance at maximum input power than the compared x86-based server.

For more information on SAP HANA running on IBM Power see `https://www.ibm.com/power/sap-hana.`

Red Hat OpenShift Container Platform

Red Hat OpenShift is the industry's leading enterprise-ready Kubernetes platform that can run anywhere – either on-premise in your data center, on IBM Cloud, or on third-party cloud providers like AWS, Azure or Google and on ppc64le, s390x, x86_64 or AMD platforms.

Red Hat OpenShift is optimized to improve developer productivity and promote innovation; it is fully supported on all IBM Power servers starting with IBM POWER9 processors when running RHCOS 4.14 or later.

For an even more flexible solution Red Hat OpenShift can be paired with Red Hat OpenShift Data Foundation or IBM Storage Fusion to provide a flexible software-defined storage solution to simplify cloud transformation projects.

1.4.2 Power architecture benefits

Application modernization comes in many shapes and sizes, and it is not always easy to know where to start. Here we describe how IBM Power brings strengths and benefits to your modernization efforts. There are many more benefits than those enumerated here. IBM Power is built for core enterprise applications and the next wave of digital transformation fueled by application modernization. Here are a few advantages of modernizing with IBM Power10.

Pervasive security and resiliency

To meet today's security challenges, it is essential that every layer of your company's IT hardware and software stack remains secured. IBM Power customers utilize the most reliable

[11] 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

mainstream server platform to innovate and get to market faster without compromising security.

IBM Power's multi-layered approach to security gives you full visibility of your hardware and software. With IBM Power10's hardware-accelerated transparent memory encryption, quantum-safe cryptography and fully homomorphic encryption, your data is protected with comprehensive end-to-end security at every layer of the stack – for both today's and tomorrow's threats.

More performance from software with fewer servers

You can buy fewer IBM Power servers to run an equivalent set of applications at comparable throughput levels than on competing platforms. That is because it provides 55% lower 3-year total cost of ownership (TCO) to run modern cloud-native applications – achieving 4.4X better per-core throughput.[12] This allows the collocation of cloud-native apps with existing AIX, IBM i and Linux virtual machine-based applications enabling access to low-latency API connections to business-critical data. Plus, you can leverage sub-capacity licensing to greatly reduce containerized software license costs (IBM Cloud Paks, for example) using PowerVM shared processor pools, allowing CPU cores to be autonomously shared across Red Hat OpenShift worker nodes without sacrificing application performance.

Superior performance for your enterprise data

Running Red Hat OpenShift in a virtual machine adjacent to your AIX, IBM i or Linux virtual machines provides low-latency reliable communication to your enterprise data with PowerVM Virtual I/O Server. This provides improved performance due to fewer network hops. It also allows for security-enhanced communication between the new cloud-native apps and your enterprise data stores as the network traffic never has to leave the physical server.

Flexible, efficient utilization

You can manage spikes in demand and support more cloud workloads per server with IBM PowerVM hypervisor on-demand CPU capacity for IBM Power compute and memory. Power virtualization technology manages demand by sharing pools of CPU cores across nodes. These differentiating hypervisor and consumption constructs – such as uncapped processors and shared processor pools – provide the ability to guarantee performance SLAs while donating unused processor cycles to worker nodes in need of additional capacity. Additionally, on-premise pay-as-you-go consumption is available for Red Hat OpenShift running on IBM Power.

Incremental application modernization

With IBM Power10, teams can incrementally modernize their existing AIX, IBM i, and Linux applications by extending them with new cloud-native services in a safe and methodical manner. This means you can capitalize on existing investments in applications and skills and drive incremental transformation – saving money, expediting time-to-value and minimizing risk.

For IBM i clients, this is made even easier with Merlin (IBM i Modernization Engine for Lifecycle Integration). Merlin is a set of tools that run in Red Hat OpenShift containers that guide and assist developers in the modernization of IBM i apps.

[12] 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.

Innovate with an extensive container software ecosystem

At the heart of any application modernization effort is a strong software ecosystem that allows teams to innovate using the latest technologies. Now, more than ever, open-source communities are playing a significant role in an organization’s modernization journeys.

IBM Power not only runs your core business applications, but also a wide range of popular open-source and commercial container software running on Red Hat OpenShift. When you choose IBM Power to modernize, you choose industry-leading reliability, performance and security, as well as superior compute performance for data-intensive and mission-critical applications. It is a foundation for modern container-based applications.

Trusted and proven foundation

Kubernetes provides the core foundation for modernizing your enterprise applications. As the industry’s leading enterprise Kubernetes platform, Red Hat OpenShift provides a consistent foundation for application development and containerized workloads to support hybrid cloud, multicloud and edge deployments. This benefits both developers and IT administrators. Your developers have access to the latest software innovations within Red Hat OpenShift to build solutions faster while your IT administrators can easily observe, operate and manage the platform and infrastructure. This helps you deliver high-value, high-quality software faster to end users. All of this is enabled through Red Hat OpenShift.

Red Hat OpenShift Container Platform

Red Hat OpenShift is the industry’s leading enterprise-ready Kubernetes platform that can run anywhere – either on-premise in your data center, on IBM Cloud, or on third-party cloud providers like AWS, Azure or Google.

Red Hat OpenShift is optimized to improve developer productivity and promote innovation; it is fully supported on all IBM Power servers starting with IBM POWER9 processors when running RHCOS 4.14 or later.

For an even more flexible solution Red Hat OpenShift can be paired with Red Hat OpenShift Data Foundation or IBM Storage Fusion to provide a flexible software-defined storage solution to simplify cloud transformation projects.

IBM Cloud Paks and Red Hat software

IBM Power provides superior performance and economics for containerized workloads like IBM Cloud Paks and an extensive set of Red Hat open-source software solutions for modernizing existing applications and developing new cloud-native apps that run on Red Hat OpenShift.

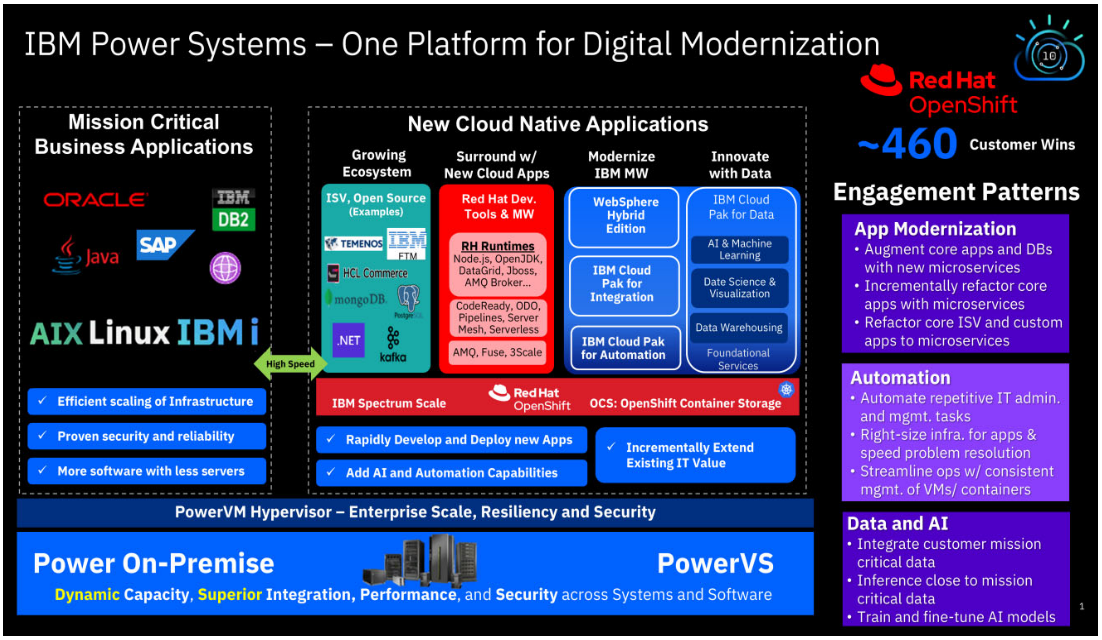
There are three main benefits of IBM Cloud Paks:

- They are comprehensive and easy to use.
- They are supported by IBM.
- They run anywhere Red Hat OpenShift runs.

IBM Cloud Paks take a bundled approach that allows you to accelerate your modernization journey by packaging everything you need to get started. The IBM Cloud Paks available on IBM Power include IBM Cloud Pak® for Applications, IBM Cloud Pak for Data, IBM Cloud Pak for Watson AIOps (Infrastructure Automation), IBM Cloud Pak for Integration and IBM Cloud Pak for Business Automation. With the addition of multiple architecture clusters it is now possible to integrate additional Cloud Pak capabilities into your IBM Power based Red Hat OpenShift clusters utilizing x86 worker nodes.

From a Red Hat software perspective, there is also a comprehensive set of software solutions to accelerate your modernization efforts, including Red Hat Runtimes, Red Hat 3scale API

Management, Red Hat Fuse and Red Hat AMQ. Figure 1-12 shows the strong portfolio available for use in modernizing applications on IBM Power.



IBM Power Systems – One Platform for Digital Modernization

Red Hat OpenShift

~460 Customer Wins

Mission Critical Business Applications

AIX Linux IBM i

- ✓ Efficient scaling of Infrastructure
- ✓ Proven security and reliability
- ✓ More software with less servers

New Cloud Native Applications

Growing Ecosystem	Surround w/ New Cloud Apps	Modernize IBM MW	Innovate with Data
ISV, Open Source (Examples)	Red Hat Dev. Tools & MW **RH Runtimes** Node.js, OpenJDK, DataGrid, Jboss, AMQ Broker… CodeReady, ODO, Pipelines, Server Mesh, Serverless AMQ, Fuse, 3Scale	WebSphere Hybrid Edition IBM Cloud Pak for Integration IBM Cloud Pak for Automation	IBM Cloud Pak for Data AI & Machine Learning Date Science & Visualization Data Warehousing Foundational Services

IBM Spectrum Scale — Red Hat OpenShift — OCS: OpenShift Container Storage

- ✓ Rapidly Develop and Deploy new Apps
- ✓ Add AI and Automation Capabilities
- ✓ Incrementally Extend Existing IT Value

PowerVM Hypervisor – Enterprise Scale, Resiliency and Security

Power On-Premise PowerVS

Dynamic Capacity, Superior Integration, Performance, and Security across Systems and Software

Engagement Patterns

App Modernization
- Augment core apps and DBs with new microservices
- Incrementally refactor core apps with microservices
- Refactor core ISV and custom apps to microservices

Automation
- Automate repetitive IT admin. and mgmt. tasks
- Right-size infra. for apps & speed problem resolution
- Streamline ops w/ consistent mgmt. of VMs/ containers

Data and AI
- Integrate customer mission critical data
- Inference close to mission critical data
- Train and fine-tune AI models

Figure 1-12 Modernization portfolio

Comprehensive hybrid cloud management and automation

As teams increasingly shift to a hybrid cloud IT model, the need for consistent management, observability and automation approaches is paramount. Consistency across hardware platforms, clouds and operating systems is crucial for IT administrators and developers.

IBM and Red Hat check these boxes with IBM Cloud Pak for Watson AIOps (Infrastructure Automation), IBM Instana Observability, IBM Turbonomic® Application Resource Management, Red Hat Advanced Cluster Management for Kubernetes and Red Hat Ansible Automation Platform — all of which extend the value of the IBM Power platform. IBM Power makes operating and automating your hybrid cloud much easier.

1.4.3 Key benefits of IBM Power compared to x86 servers

It is a common belief that x86 servers offer the best foundation for cloud computing, largely due to their perceived low cost. While x86 systems often have lower upfront acquisition costs, this perspective overlooks critical factors such as performance, scalability, energy efficiency, data center space utilization, reliability, and overall manageability. When these elements are considered, the total cost of ownership (TCO) and return on investment (ROI) for x86 servers typically fall short compared to IBM Power Systems. IBM Power offers a more robust and cost-effective platform, delivering superior value through its unique combination of performance and enterprise-grade features:

- SAP SD-two tier benchmark results with 8 sockets (120 cores), beating the 16 socket (448 cores) result from the x86 platform.[13]
- Power delivers per-core performance that is 2.5X faster than Intel Xeon Platinum.[14]
- When running containerized applications and databases on an IBM Power E1080 compared to running the same workloads on an x86 server, IBM Power delivers 48%

[13] https://www.sap.com/dmc/exp/2018-benchmark-directory/#/sd

[14] https://www.spec.org/cpu2017/results/cpu2017

lower 3-year TCO, 4.3X more throughput per core, and 4.1X better price-performance. This means you can run the same amount of workloads with fewer servers, four times less footprint, four times fewer software licenses, and four times the energy savings.[15]

- In an ITIC 2023 Global Server Hardware, Server OS Reliability Survey, IBM Power delivered the top reliability results for the 15th straight year, better than any Intel x86 platform, and only exceeded by the IBM Z. IBM Power also reported less number of data breaches (one) in the same period compared to x86 platforms.[16]
- As shown by the list of supported operating systems in Table 1-1 on page 12, IBM Power delivers the ability to run a wide variety of AIX, IBM i, or Linux workloads simultaneously, giving you flexibility in virtualization that is unmatched by any x86 offering.

Both IBM Power and x86 architectures are established, mature foundations for modern workloads, but Power stands out for its efficiency, deeply integrated virtualization, highly dependable availability and reliability, and its unparalleled scalability. This provides the ability to support enterprise-class workloads with significantly less infrastructure compared with what is required for running on x86 hardware.

1.5 Modernization components

Modernization upgrades IT infrastructure, applications, and workflows for better agility, efficiency, and scalability. This section covers key areas such as application modernization, containerization, and automation.

1.5.1 Application modernization

Application modernization refers to the transformation of legacy applications to leverage contemporary architectures, improving maintainability, scalability, and interoperability. The process includes rehosting, replatforming, refactoring, or replacing applications based on business and technical requirements.

Understanding application types is critical for defining the appropriate modernization strategy. Applications typically fall into the following categories:

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

We discuss each of these types in the following sections.

Traditional (monolithic)

Monolithic applications are characterized by tightly coupled components within a single, self-contained architecture. This model has historically been the foundation of enterprise IT but presents challenges in scalability, maintainability, and agility. Monolithic applications are a single, indivisible unit, where all the components are tightly combined and packed together, usually running inside a virtual machine, also known as a logical partition (LPAR).

[15] https://www.ibm.com/it-infrastructure/resources/power-performance/e1080/#5
[16] https://itic-corp.com/itic-2023-reliability-survey-ibm-z-results/

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

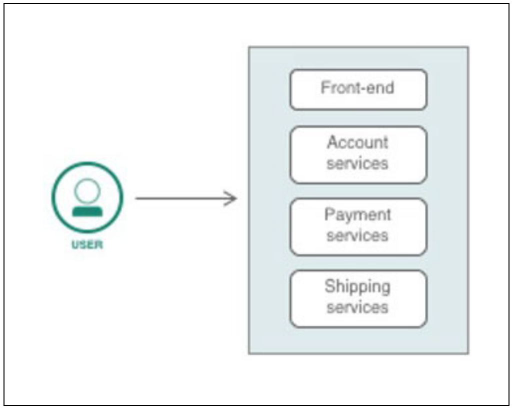


Figure 1-13 Monolithic application model

Monolithic application contains all required layers for an application to operate:

- User interface (UI)

  The UI manages what is seen by the user, including images, text and anything else that can be transmitted over the UI screen

- Business logic

  Business logic is the part of the application that encodes the business rules that determine how data can be manipulated.

- Data access layer

  The data access layer provides simplified access to data stored in persistent storage.

- Application Integration layer

  The application integration layer is responsible for integration with other services or data sources.

Monolithic applications feature centralized architecture with interconnected components. They contain complex dependencies between business logic, user interface, and data storage and provide limited adaptability for cloud-based or distributed environments.The also are characterized by the use of a waterfall development methodology with predefined release cycles.

Figure 1-14 illustrates these challenges, emphasizing the architectural barriers that traditional monolithic applications face in adapting to modern IT environments.

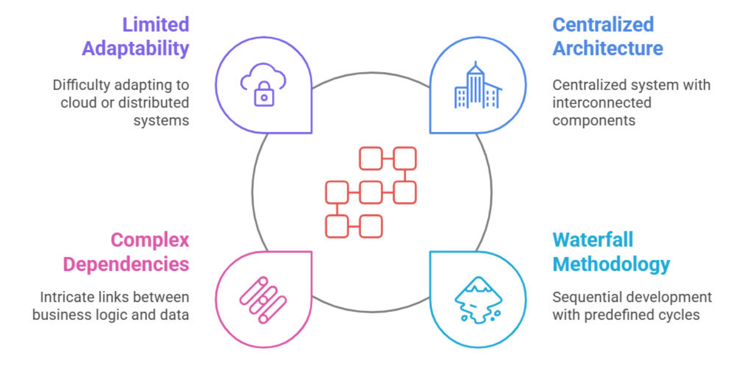

Limited Adaptability
Difficulty adapting to cloud or distributed systems
Centralized Architecture
Centralized system with interconnected components
Complex Dependencies
Intricate links between business logic and data
Waterfall Methodology
Sequential development with predefined cycles

Figure 1-14 Challenges of monolithic applications

Key characteristics of a monolithic applications

Key characteristics of a monolithic application are:

- Single Codebase

  In monolithic software, all of the code required for an application is kept in one central location.This provides an added simplifying benefit for development as communication happens in one format and work is done in shared environment.

- Tightly combined

  All components are interconnected. Any changes made to one part of the application can have unintended consequences on other parts or the entire application.

- Self-Contained

  Traditional applications are designed to work independently.

Advantages and disadvantages of monolithic applications

The advantages of monolithic applications are:

- Easier development

  Constructed with one codebase applications are more straightforward especially for smaller projects with well-defined requirements.

- Deployment simplicity

  Monolithic applications are typically deployed as single unit, making the integration process less complex.

- Testing and debugging

  A single codebase simplifies testing and debugging for small to medium-sized applications due to reduced complexity and streamlined processes. However, large and complex applications with a single codebase can make these tasks time-consuming due to increased interdependencies and complexity.

- More cyberthreat proof

  With an architecture of a closed system, access to all activities and data processing is more limited, hence more protected.

Some disadvantages of cloud native applications are:

- Limited agility

  The tightly coupled components of a monolithic application limits the possibility of introducing changes or implementing new features without major refactoring.Changes in one area can affect the entire application.

- Reduced scalability

  Scalability is the greatest challenge monolithic architectures face. Even if the amount of scaling needed is relatively minor (like adjusting a single function) you may have to effectively dismantle the system and rebuild it so it reflects the new change. That can prove time-consuming and labor-intensive.

- Resistant to other technologies

  Monolithic applications are often limited to a single technology stack.

Cloud native

In a cloud environment, the demand for new functionalities, runtime integration, and constant resource scaling, along with the necessity for applications to be deployable on new LPARs or containers, are just a few of the requirements that monolithic applications struggle to meet.

This is due to their inherent development nature, which complicates maintenance and updates.

Cloud-native applications are designed to be modular, scalable, and agile, leveraging containerization, microservices, and API-driven integration. These applications align with DevOps methodologies to enable continuous deployment and rapid adaptation to business needs.

Cloud-native applications represent the evolution from legacy monolithic architectures to agile, scalable, and API-driven solutions. The transition follows structured modernization stages, including modular design, API integration, cloud deployment, and continuous integration/continuous deployment (CI/CD) automation. These elements enable businesses to adopt cloud-agnostic platforms, enhance system interoperability, and support modern development frameworks.

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

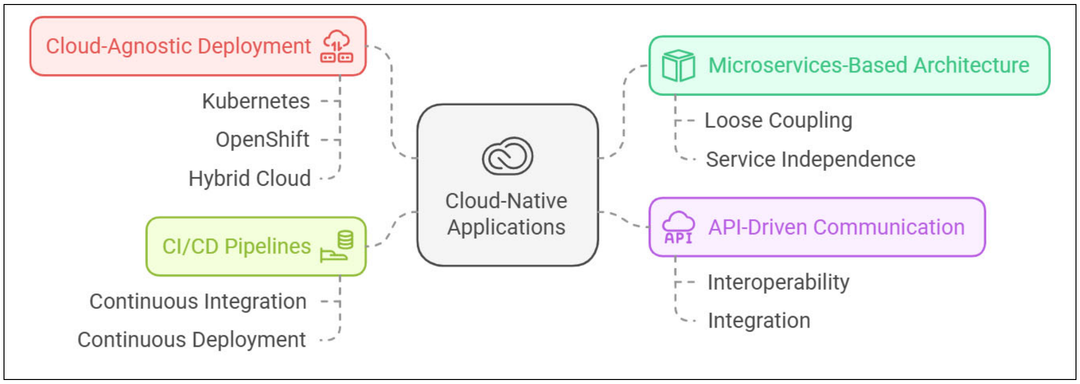


Figure 1-15 Cloud-native application architecture

Cloud native architecture, often called microservices architecture, is an approach in which a single application is built using smaller blocks of code which are independently deployable instead of being one monolithic block. The smaller components are often called services, Every function of an application is deployed as a service. This is shown in Figure 1-16.

.

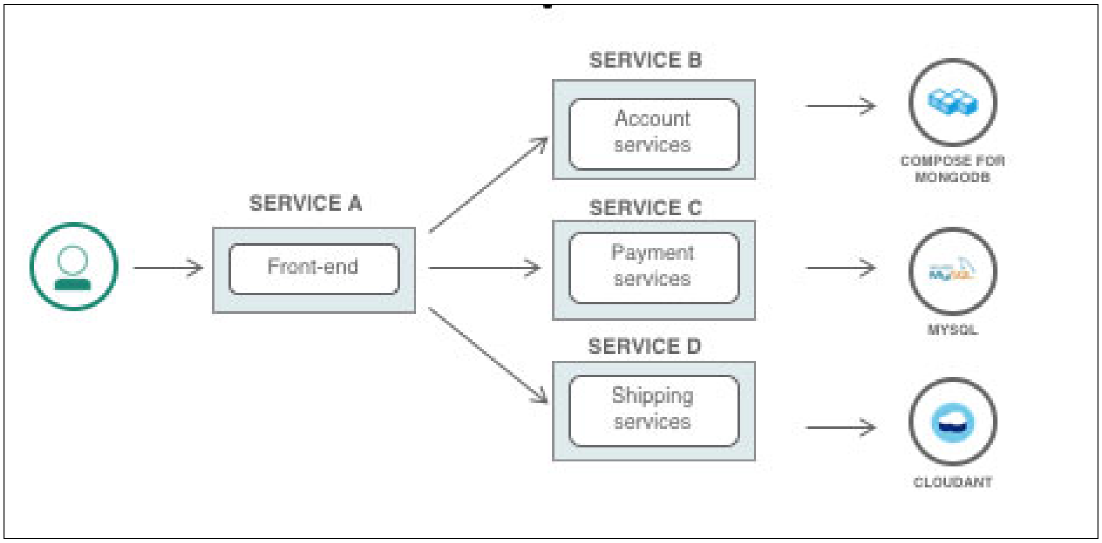


Figure 1-16 Microservices or cloud native model

Key characteristics of a cloud native applications

Key characteristics of a cloud native application are:

- Decentralized ownership

  Often development is spread between multiple teams, each responsible for a specific microservice

- Independent Deployment

  Microservices can be updated, tested, scaled and updated independently of one another.

- Service-to-service communication

  All microservices inside an application interact to one another via lightweight protocols and APIs.

- Function separation

  For each well defined function there is a microservice in place.

Advantages and disadvantages of cloud native applications

The advantages of cloud native applications are:

- Accelerate scalability

  Better suited for handling large applications.Multiple services can scale simultaneously.

- Geared for automation

  Microservices allow organizations to automate the continuous integration/continuous delivery (CI/CD) process. This enables development of code updates that occur on a continuing schedule.

- Quicker deployment

  Due to the nature of the application it is much quicker to deploy a single service, rather then the entire application.Services are deployed independently.

- Increased cost efficiency

  optimized resource allocation and maintenance due to development happens on well-defined services. Efforts are localized to specific service, reducing overall costs.

Some disadvantages of cloud native applications are:

- Increased complexity

  As microservices are distributed, managing service communication can be challenging. Development may need to write extra code to ensure smooth communication between modules.

- Vulnerable security

  Service-to-service communication uses an API gateway. This can create a security exposure when it comes to authentication and other critical processes.

Composite

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

Composite applications provide integration of existing enterprise applications with modern APIs allowing partial migration to cloud environments while retaining some legacy components. This is accomplished through the use of middleware and API gateways to bridge communication between monolithic and cloud-native architectures, allowing enhanced agility through modular refactoring strategies.

Composite applications can combine several different sources, applications, services and even include an entire application whose outputs are packed in a single new application presented to the end user. In other words, they create modernized front ends for legacy systems. This allows them to be built on any technology or architecture. This is shown in Figure 1-17.

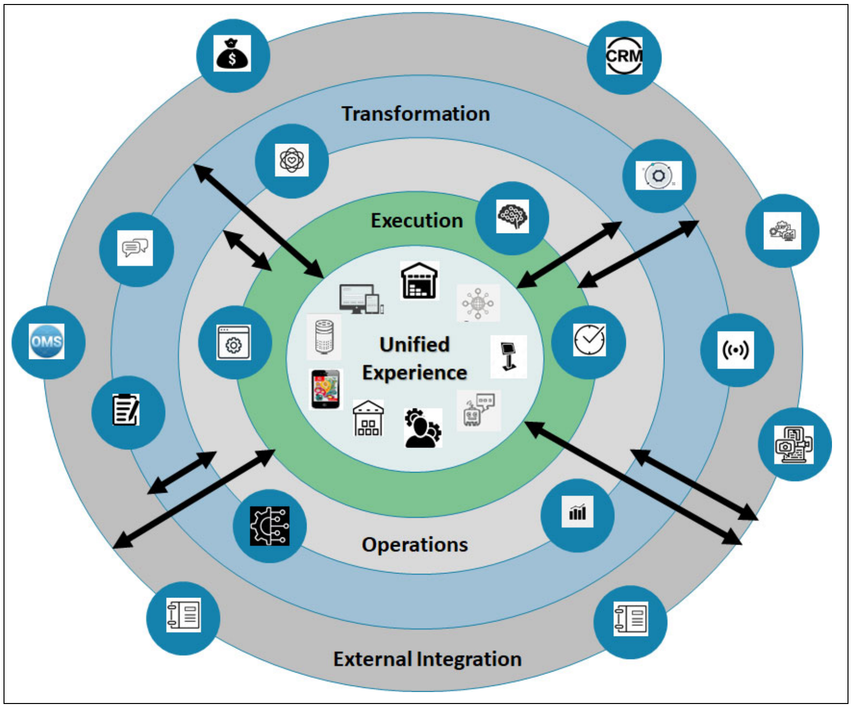


Figure 1-17   Composite application model

Key characteristics of a composite applications

Key characteristics of a cloud native application are:

- Modular Design:

  Composite applications are built by integrating various components (like web services, databases, and applications) which can be developed and managed independently.

- External Service Integration:

  They heavily rely on accessing data and functionalities from external services.

- Orchestration Layer:

  A middleware layer is often needed to manage the interactions and data flow between different components, ensuring seamless integration and coordinated execution of tasks.

Advantages and disadvantages of composite applications

The advantages of composite applications are:

- Technology diversity:

  Different technology stacks can be incorporated during the building process of a composite application.

- Unified User Interface:

Presents data from various systems in a single, user-friendly interface, eliminating the need to navigate multiple applications to access necessary information.

- Improved Data Integration:

  Enables seamless access to data from different sources, including legacy systems, without requiring complex data transformation or manual data entry.

- Customizable Applications:

  Enables creation of tailored applications by combining different services to meet specific business requirements

Some disadvantages of cloud native applications are:

- Domino Effect:

  If an underlying software goes down, the application may not work properly as the it relies heavily on the availability and functionality of this software.

- Security considerations:

  Managing security across multiple integrated systems within a composite application can be complex and requires careful planning.

- Troubleshooting challenges:

  Identifying the root cause of an issue inside a composite application can be difficult due to the component interconnection architecture.

1.5.2 Virtualization technologies

Virtualization is a process whereby software is used to create an abstraction layer over computer hardware that allows the hardware elements of a single computer (or host machine) to be divided into multiple virtual computers.

A hypervisor is a small layer that enables multiple operating systems to run alongside each other, sharing the same physical computing resources. A hypervisor allows the physical computer to separate its operating system and applications from its physical hardware.

A hypervisor is a software that enables multiple Virtual Machines (VMs) with each VM running their own Operating System (OS) to run on one physical server. The hypervisor pools and allocates physical computing resources as needed by the VM, enabling efficiency, flexibility and scalability. Hypervisors separates VMs from each other logically, assigning each its own slice of the underlying computing power, memory and storage. This prevents the VMs from interfering with each other. For example, if one OS suffers a crash or a security compromise, the others survive.

Simply put, a virtual machine is an emulation of a physical computer. VMs enable teams to run what appear to be multiple machines, with multiple operating systems, on a single computer. VMs interact with physical computers by using lightweight software layers called hypervisors.

Containerization compared with virtualization

Containers have become a dominant force in cloud native applications so it's important to understand what they are and what they are not. While containers and Virtual Machines have distinct and unique characteristics, they are similar in that they both improve efficiency, application portability, and enhance the software development lifecycle.

Containers and Virtual Machines are two approaches to packaging computing environments that isolate them from the rest of the system. The main difference between the two is what components are isolated, which in turn affects the scale and portability of each approach. Containers are a lighter-weight, more agile way of handling virtualization because they don't use a hypervisor. Rather than spinning up an entire virtual machine, containers packages together everything needed to run a single application or microservice (along with runtime libraries they need to run).

Containers use a form of operating system (OS) virtualization. Containers leverage features of the host operating system to isolate processes and control the processes' access to CPUs, memory and disk size.

In most operating systems, code runs in either user space or kernel space. Code executing in kernel space has direct, unrestricted access to all system hardware. As a result, a crash in kernel space can render the entire system unavailable, while a crash in user space typically affects only the specific application.

A container emulates the user space of an operating system. Through container engines like 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, but all containers share the same underlying hardware and operating system kernel.

This architecture offers a secure and lightweight application environment. Containers start up almost instantly because they don't require booting a full OS kernel—the host kernel is already running. However, this design also introduces a single point of failure, since all containers rely on the same kernel and hardware. This risk is mitigated through availability, scaling, and redundancy mechanisms provided by container orchestration platforms like Red Hat OpenShift Container Platform.

As a result, containers are ideally suited for microservices and stateless applications, where lightweight, scalable deployment is key. In contrast, critical stateful applications, such as databases, are often better deployed in virtual machines (VMs), which provide dedicated hardware abstraction and kernel isolation, ensuring stronger fault isolation and resource control.

Containers have been around for decades. However, the common consensus is that the modern container era began in 2013 with the introduction of Docker, an open source platform for building, deploying and managing containerized applications.

Instead of virtualizing the underlying hardware, containers virtualize the operating system so each individual container contains only the application and its libraries and dependencies. Containers are small, fast, and portable because they do not need to include a guest OS in every instance because they leverage the features and resources of the host OS.

As part of a modernization journey on IBM Power, containers are generally 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 systems.

PowerVM

IBM PowerVM is the virtualization hypervisor that comes standard on IBM Power, IBM PowerVM provides support for virtual machines, enabling the creation of logical partitions (LPARs) and supports sharing of resources across multiple partitions. PowerVM is tightly integrated to the IBM Power hardware providing enterprise grade virtualization with minimal overhead compared to other virtualization technologies. PowerVM allows you to consolidate VMs running multiple workloads onto fewer systems, resulting in reduced costs, increased efficiency, better return on investment, faster deployment, workload security, and better server utilization.

The PowerVM Hypervisor (PHYP) is the built-in hypervisor for IBM Power Servers which is embedded in the system firmware. When the Power system is switched on, the PHYP is loaded together with the system firmware. This minimizes the virtualization overhead for guest operating systems, and provides the following Hypervisor capabilities;

- IBM Micro-Partitioning®.
- Shared Processor Virtualization Pool (SPP)
- Virtual I/O Server (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, which enables support for 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. Ubuntu is currently only available on bare metal Power systems.

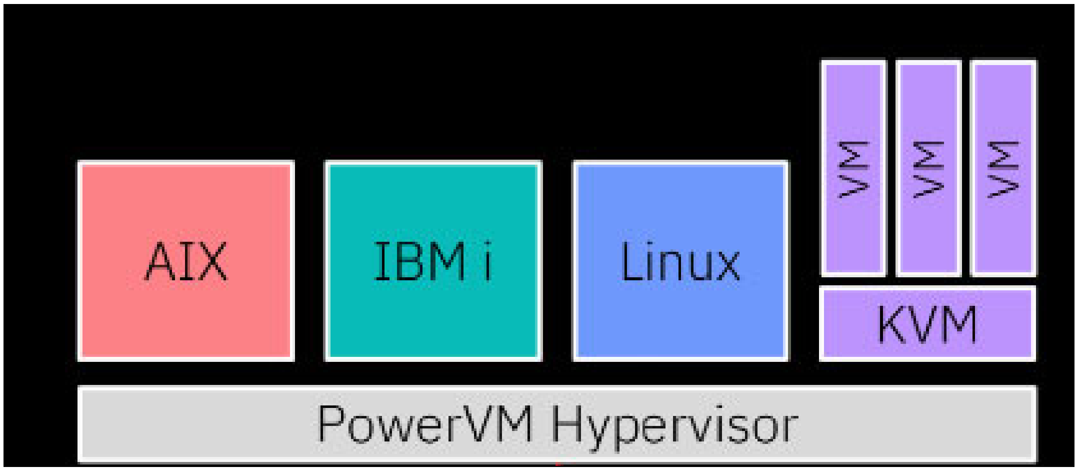


Figure 1-18 Supported guest operating systems on IBM PowerVM Hypervisor

PowerVM enables a Power server to have up to 1000 virtual machines on a single server running a mix of various operating systems and environments simultaneously. PowerVM also provides IBM Power other advanced features to help you manage and control your virtualized environment.

PowerVM enables virtualization of the hardware, from processor to memory and storage I/O to network I/O resources. It enables platform-level capabilities like Live Partition Mobility (LPM) or Simplified Remote Restart (SRR)

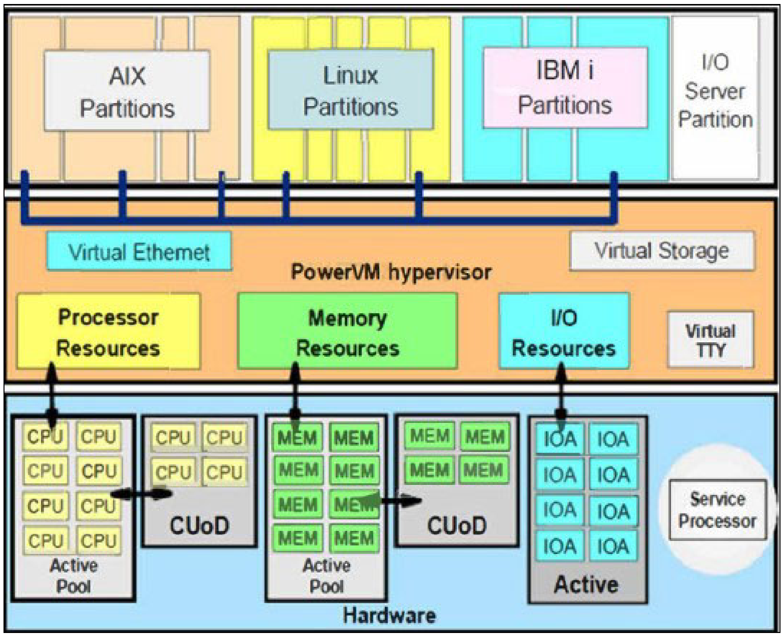


Figure 1-19 PowerVM Overview

Micro-partitioning

PowerVM allows a VM to initially occupy as small as 0.05 processing units, or 1/20 of a single processor core, and allows adjustments as small as a hundredth (0.001) of a processor core. This allows tremendous flexibility in the ability to adjust your resources according to the exact needs of your workload.

In PowerVM with micro-partitioning, you can not only see the actual granularity of processor allocation to a VM by as little as a hundredth (0.001) of the CPU, but can also provision CPU resources from targeted processor pools.

PowerVM micro-partitioning delivers immense flexibility and efficiency in allocating valuable assets. Ensuring CPU allocations are not unnecessarily wasted on larger than needed VMs.

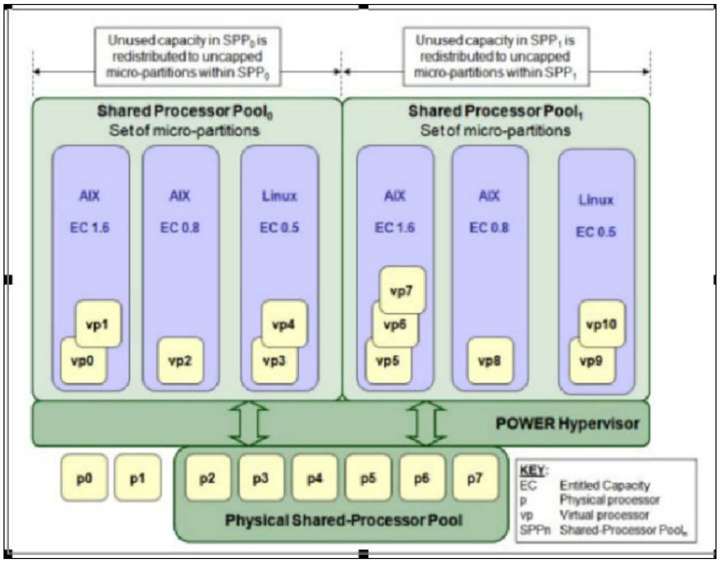


Figure 1-20 Overview of the architecture of multiple shared pools[17]

Shared Processor Pools

Allows for effective overall utilization of system resources by automatically applying only the required amount of processor resource needed by each partition. The hypervisor can automatically and continually adjust the amount of processing capacity allocated to each partition/VM based on system demand. You can set a shared processor partition so that, if a VM requires more processing capacity than its assigned number of processing units, the VM can use unused processing units from the shared processor pool.

Shared processors are physical processors whose processing capacity is shared among multiple logical partitions. The ability to divide physical processors and share them among multiple logical partitions is known as the Micro-Partitioning technology.

The server distributes unused capacity among all of the uncapped shared processor partitions that are configured on the server, regardless of the shared processor pools to which they are assigned. For example, if you configure logical partition 1 to the default shared processor pool and you configure logical partitions 2 and 3 to a different shared processor pool, all three logical partitions compete for the same unused physical processor capacity in the server, even though they belong to different shared processor pools.

> **Note:** To read more about shared processors please visit.
> https://www.ibm.com/docs/en/power10?topic=processors-shared

Virtual I/O Server

The Virtual I/O Server (VIOS) provides the ability to share storage and network resources across several VMs simultaneously, thereby avoiding excessive costs by configuring the precise amount of hardware resources needed by the system.

The VIOS is a software that is located in a logical partition, and a required part of the PowerVM Editions hardware feature to provision sharing of physical I/O resources between client logical partitions within the IBM Power Managed System. The VIOS provides virtual Small Computer Serial Interface (SCSI) target, virtual Fibre Channel, and Shared Ethernet Adapter to client logical partitions within the system.

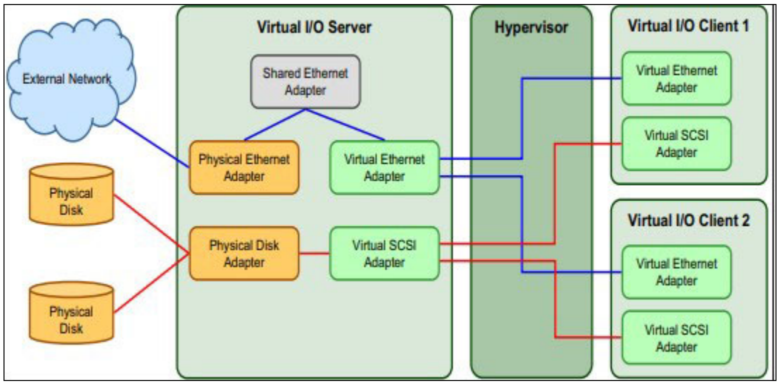


Figure 1-21 Architecture view of the VIOS

The recommended best practice for deployment of the VIOS is in a dual VIOS configuration on each IBM Power Managed System.

> **Note:** You can find more information about the VIOS from this location.
> https://www.ibm.com/docs/en/power10/9080-HEX?topic=server-virtual-io-overview

[17] https://www.redbooks.ibm.com/technotes/tips1091.pdf

Virtual I/O Server Storage Support

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

- Virtual SCSI (vSCSI)

  When using vSCSI the storage adapters and volumes are assigned to the VIOS server and virtual SCSI adapters are defined connecting those devices to the client logical partitions. These vSCSI can share disk storage, tape or optical devices that are assigned to the Virtual I/O Server (VIOS) logical partition. All storage device requirements remain on the VIOS LPARs, simplifying the configuration of the client LPARs.

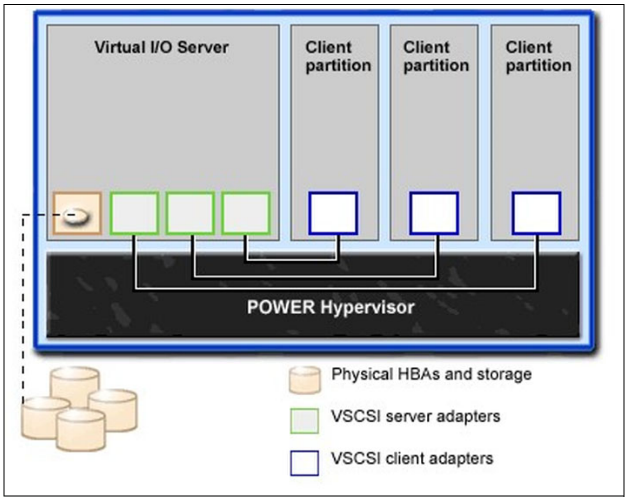


Figure 1-22 Standard vSCSI configuration for a Managed System

Note: For a better overview of vSCSI storage please visit the following URL:
`https://www.ibm.com/docs/en/power10?topic=overview-virtual-scsi`

- Shared Storage Pool (SSP)

  The shared storage pool is an extension of the vSCSI connection in which the connected storage devices assigned to the VIOS are configured into a shared storage pool. which can further simplify storage management,

  A shared storage pool is a collection of storage resources that can be shared across multiple virtual machines or partitions in a PowerVM environment. It provides a unified, flexible storage infrastructure for virtualized systems. The storage pool abstracts physical storage resources, so the guest operating systems within LPARs do not need to be aware of the underlying physical storage infrastructure. Each VM can access the storage resources in the pool based on defined policies, such as how much storage to allocate or how to manage data replication or redundancy.

  Multiple physical disks can be pooled together to create one virtualized storage system, abstracting away individual disk management. VMs can dynamically request and release storage from the pool based on their needs. This helps in efficient utilization of available resources. The Virtual I/O Server (VIOS) manages these pools, allowing partitions (virtual machines or logical partitions - LPARs) to access the storage transparently.

  If a disk or storage unit fails, the pool can provide failover mechanisms, ensuring that virtual machines continue to function without significant downtime. Storage in the pool can be resized, expanded, or contracted, making it adaptable to changing workload requirements.

When using SPP storage is more efficiently used because it can be dynamically allocated to where it's needed, avoiding wasted capacity. New storage can be added to the pool without downtime, allowing the system to grow with increasing demands and storage that is not in use by one VM can be easily reassigned to others, improving overall system performance and utilization.

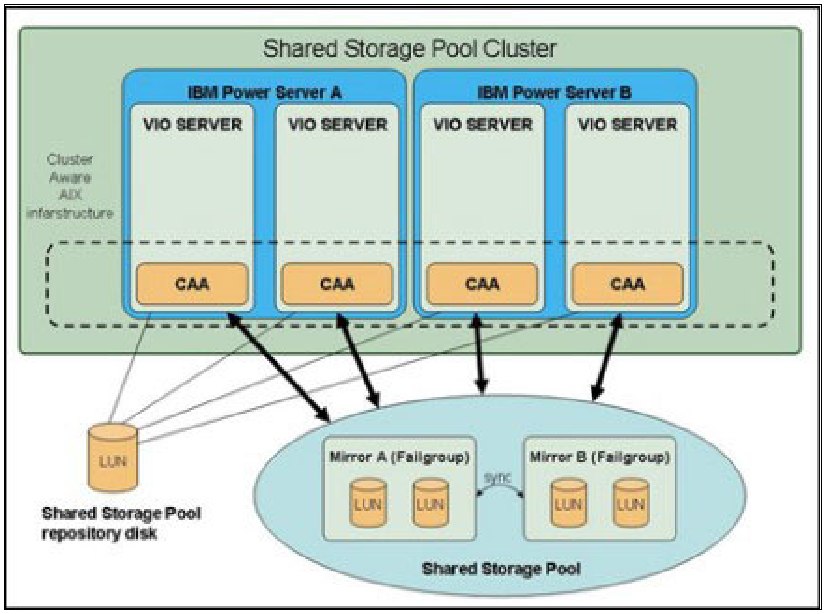


Figure 1-23 Standard SSP architecture

- N_Port ID Virtualization (NPIV)

  Using NPIV allows virtual Fibre Channel devices to pass through the VIOS directly to the partition using virtual Fibre Channel adapters. The LUNs are zoned directly from the SAN to the partition's virtual Fibre Channel adapter. This technology allows multiple logical partitions to access independent physical storage through the same physical Fibre Channel adapter port on the VIOS while maintaining the security of zoning in the Fibre Channel. This is shown in Figure 1-24.

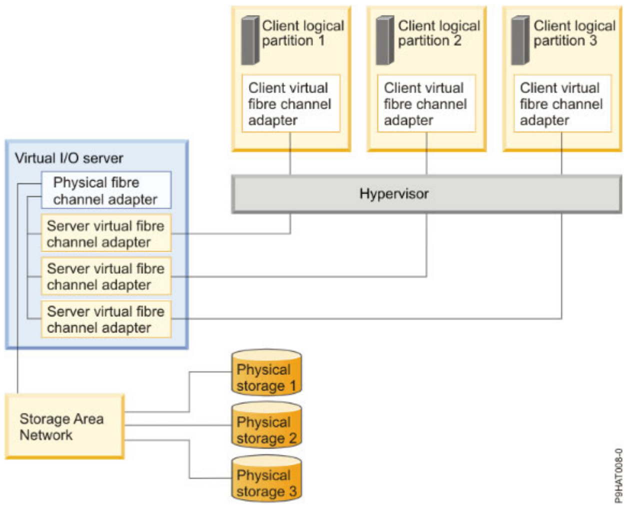
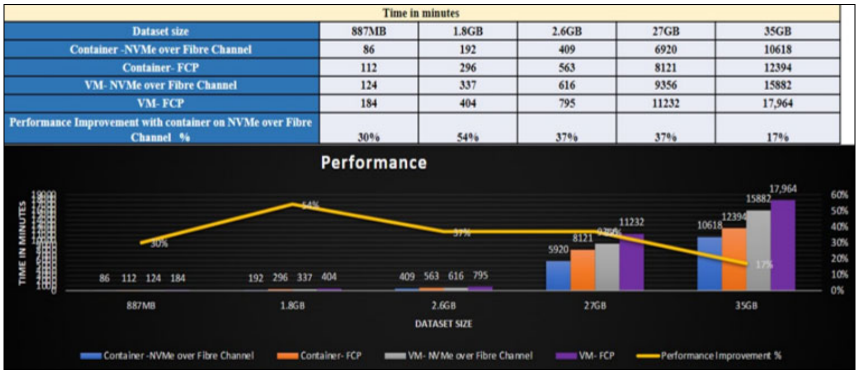

Client logical partition 1
Client logical partition 2
Client logical partition 3
Client virtual fibre channel adapter
Client virtual fibre channel adapter
Client virtual fibre channel adapter
Virtual I/O server
Physical fibre channel adapter
Server virtual fibre channel adapter
Server virtual fibre channel adapter
Server virtual fibre channel adapter
Hypervisor
Physical storage 1
Physical storage 2
Physical storage 3
Storage Area Network
P9HAT008-0

Figure 1-24 NPIV for Fiber Channel device support

For production environments, IBM recommends using Fibre Channel devices with Network Port Virtualization (NPIV) in a dual VIOS configuration for SAN Storage volume redundancy.

- Fibre Channel (FC)-Nonvolatile Memory express (NVMe)

  New enhancements in NPIV on the Power10 provides the ability to use the Nonvolatile Memory express (NVMe) protocol over Fibre Channel fabrics. NVMe over Fibre Channel provides much faster access between hosts and storage systems. Figure 1-25 shows the performance advantages of NVMe over conventional SCSI devices.



Time in minutes					
Dataset size	887MB	1.8GB	2.6GB	27GB	35GB
Container -NVMe over Fibre Channel	86	192	409	6920	10618
Container- FCP	112	296	563	8121	12394
VM- NVMe over Fibre Channel	124	337	616	9356	15882
VM- FCP	184	404	795	11232	17,964
Performance Improvement with container on NVMe over Fibre Channel %	30%	54%	37%	37%	17%

Figure 1-25 NVMe performance

NVMe over FC has been proven to produce more efficient I/O performance for AI workloads when compared to traditional FCP and SCSI protocols. NVMe over Fibre channel enables faster access between hosts and storage systems with a lighter and leaner driver stack that runs faster and consumes less resources.

Note: Please watch the BrightTalk presentation on the Benefits of FC-NVMe for Containerized ML Models

The PowerVM Virtual Switch

PowerVM provides the ability to define virtual Ethernet adapters to allow client partitions to send and receive network traffic without a dedicated physical Ethernet adapter. A Virtual Ethernet adapter can also be used to allow logical partitions within the same system to communicate without having to use physical Ethernet adapters. Within the system, virtual Ethernet adapters are connected to an IEEE 802.1Q virtual Ethernet switch. The default name is ETHERNET0, but there exists the ability within PowerVM to create additional Virtual Switches based on the customers requirements.

The PowerVM Virtual switch default operation is in Virtual Ethernet Bridge (VEB) mode, and can also operate in Virtual Ethernet Port Aggregator (VEPA) mode. Using the PowerVM Virtual Switch, logical partitions can communicate with each other by using virtual Ethernet adapters and assigning tagged VIDs. With VIDs, virtual Ethernet adapters can share a common logical network. The system transmits packets by copying the packet directly from the memory of the sender logical partition to the receive buffers of the receiver logical partition without any intermediate buffering of the packet.

When the PowerVM Hypervisor switch is operating in VEB mode, VMs in the same Tagged VID can use the Hypervisor virtual switch for high performance network frame relays. When using non-bridged PVIDs the network packet does not leave the Hypervisor for in memory direct transfers.

Virtual Edge Port Aggregator, VEPA mode, part of the IEEE 802.1Qbg design to off load the complexities and performance requirements of hypervisor virtual switch bridging between many virtual machines. This mechanism pushes all frame relay switching to outside of the hypervisor virtual switch to the departmental switch network. In this configuration the departmental switch handles the frame relay for Virtual Switch guest port to guest port communications. The departmental switch is then responsible for network appliances as firewalls, Access Control Lists (ACLs), Quality of Service (QoS), and port mirroring.

When the VM to VM traffic is forced out to the Enterprise switch for firewall filtering this pushes the load of network packet management in a virtualized environment away from the hypervisor. This decreases the load on the hypervisor which can improve performance of partitions with high network utilizations.

Live Partition Mobility (LPM)

LPM brings the ability to move running VMs across different physical systems without disrupting the operating system and applications running within them.

By using Active partition migration, or Live Partition Mobility, you can migrate AIX, IBM i, and Linux logical partitions that are running, including the operating system and applications, from one system to another. The logical partition and the applications that are running on that migrated logical partition do not need to be shut down. The abilities of LPM come bundled within the PowerVM Hypervisor. Enablement is through a licensed capability, and requires a best practice implementation of PowerVM.

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

Dynamic LPAR operations (DLPAR)

Dynamic LPAR operations (DLPAR) Introduce the ability to dynamically allocate additional resources, such as available cores and memory, to a VM without stopping the application.

- Performance and Capacity Monitoring

  Supports gathering of important statistics to provide an administrator information regarding physical resource distribution among VMs and continuous monitoring of resource utilization levels, ensuring they are evenly distributed and optimally used.

- Capacity on Demand (CoD)

  When your IBM Power System is delivered it will come with processors in a combination of the below states.

  - static - delivered and dedicated to the PowerVM managed system
  - mobile - can be moved between PowerVM managed systems.
  - inactive/dark - capacity within the PowerVM managed system for future activation as either static of mobile.

  With Capacity on Demand (CoD) offerings you can dynamically activate one or more resources on your server as your business peaks dictate. You can activate inactive processor cores or memory units that are already installed on your server on a temporary and permanent basis.

Note: Learn more about capacity on demand at the following locations.

https://www.ibm.com/docs/en/power10?topic=environment-capacity-demand
https://www.ibm.com/docs/en/entitled-systems-support?topic=demand-elastic-capacity-overview-specifications

- Remote Restart

  Allows for quick recovery in your environment by allowing you to restart a VM on a different physical server when an error causes an outage.

  Simplified remote restart is a supported, configurable high availability option for logical partitions on Power10 processor-based systems. When an error causes a server outage, the simplified remote restart capability enables a restart on a different physical server. The use of capacity on demand entitlements can be used to provide the dynamic availability of system resources on the remote machines. SRR is 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 can function as a hypervisor that runs multiple, isolated virtual machines (VMs). In the KVM architecture, each guest (virtual machine) is implemented as a regular Linux process. After you install KVM, you can run multiple guests, with each of them running a different operating system image. Each of these virtual machines has private, virtualized hardware, which includes memory, storage, and a network card.

KVM in a PowerVM LPAR

Kernel-based Virtual Machine (KVM) is an extra virtualization option on Power10 systems that run on PowerVM. KVM brings the power, speed, and flexibility of the KVM virtualization technology to a PowerVM logical partition (LPAR). An LPAR that runs a KVM-enabled Linux distribution can host PPC64-LE KVM guests. The KVM Guests can use the existing resources that are assigned to the LPAR.

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

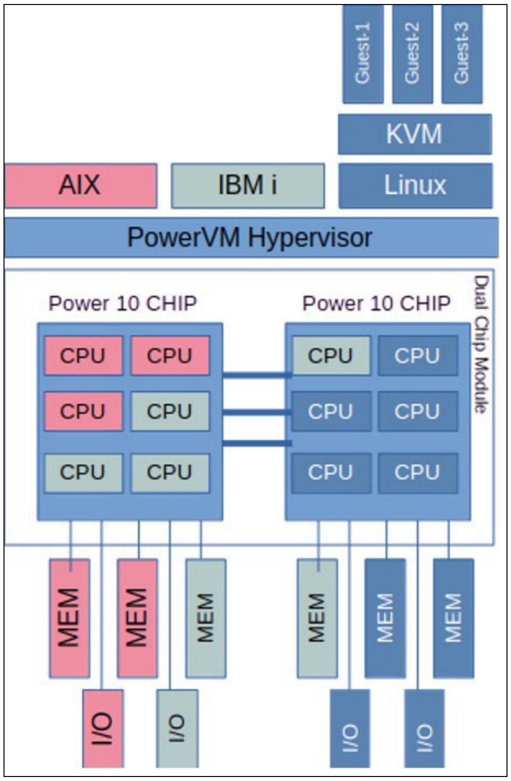


Figure 1-26 PowerVM with KVM support

Software levels

To enable KVM in a Power10 logical partition, you must meet the following code levels and distributions:

- – Firmware level: FW1060.10
- – HMC Levels: V10 R3 SP1060 or later

KVM is enabled in Kernel 6.8 and QEMU 8.2 and is working with the following Linux distributions:

- – Fedora 40 with kernel 6.10
- – Ubuntu 24.04

Industry Standard Linux Virtualization Stack

KVM in a PowerVM LPAR utilizes the industry standard Linux KVM virtualization stack and can easily integrate within an existing Linux virtualization ecosystem.

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

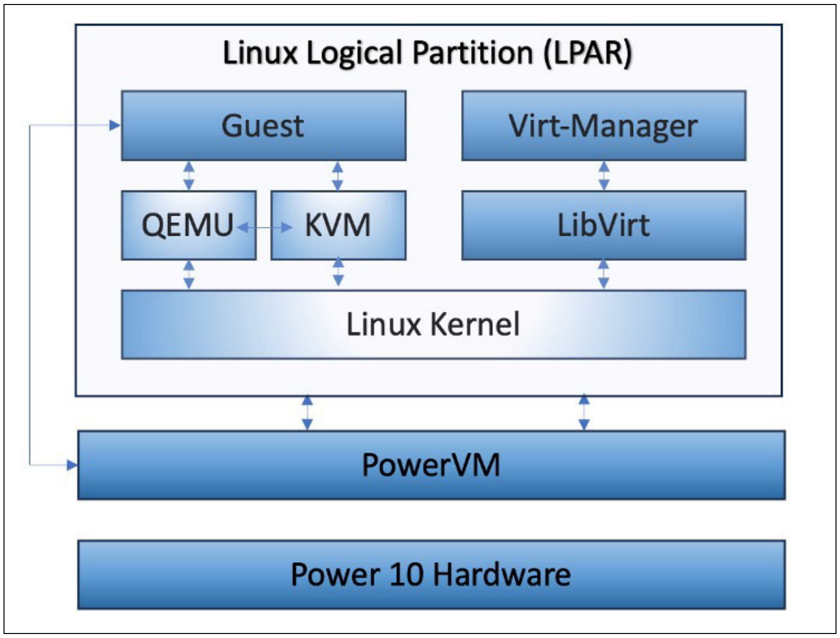


Figure 1-27 Industry Standard Linux Virtualization Stack

KVM in an LPAR is enabled by:

- IBM Power architecture and Power10

  This implementation has advanced virtualization capabilities to run multiple operating system (OS) instances that share the same hardware resources while providing isolation. The Radix MMU architecture provides the capability to independently manage page tables for the LPAR and the KVM guest instances on the LPAR.

- PowerVM

  The industry-leading virtualization stack provides new functions to create and manage KVM guests. These changes extend the Power platform architecture to include new hypervisor interfaces.

- Linux kernel that includes the KVM kernel module (KVM)

Provides core virtualization infrastructure to run multiple virtual machines in a Linux host LPAR. Upstream kernels and enabled downstream distributions such as Fedora and Ubuntu use the newly introduced Power architecture extensions to create and manage KVM guests in the Power Linux LPAR.

- QEMU

  User space component that implements virtual machines on the host that use KVM functions.

- LibVirt

  Provides a toolkit for virtual machine management.

For additional technical details refer to
`https://www.ibm.com/docs/en/linux-on-systems?topic=servers-kvm-in-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 are a lighter-weight, more agile way of handling virtualization - since they don’t use a hypervisor.

Instead of virtualizing the underlying hardware, containers virtualize the operating system (typically Linux or Windows) so each individual container contains only the application and its libraries and dependencies. Containers are small, fast, and portable because, unlike a virtual machine, containers do not need to include a guest OS in every instance and can, instead, simply leverage the features and resources of the host OS.

Just like virtual machines, containers allow developers to improve CPU and memory utilization of physical machines. However, containers go even further because they also enable microservice architectures, where application components can be deployed and scaled more granularly. This is an attractive alternative to having to scale up an entire monolithic application because a single component is struggling with load.

Kubernetes

Kubernetes, often abbreviated as K8s, is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. It provides a framework to efficiently manage the complexities of deploying and running applications in containers across a cluster of machine.

The main building blocks of a Kubernetes cluster are:

- Control Plane

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

- Worker Nodes

  Worker nodes, also known as worker machines or worker servers, are the heart of a Kubernetes cluster.

- Pods

  Pods are fundamental building blocks in Kubernetes that group one or more containers together and provide a shared environment for them to run within the same network and storage context.

- Controller

  Controllers are crucial components responsible for maintaining the desired state of resources in the cluster.

- Services

  Services are a fundamental concept that enables communication and load balancing between different sets of Pods, making your applications easily discover able and resilient.

- Volumes

  Volumes are a way to provide persistent storage to containers within Pods.

- ConfigMaps

  ConfigMaps are used to store non-sensitive configuration data as key-value pairs.

- Secrets

  Secrets are similar to ConfigMaps but are specifically designed for storing sensitive information.

- Namespaces

  Namespaces are a way to organize and partition resources within a cluster.

- Ingress

  ingress is a resource that manages external access to services within your cluster

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

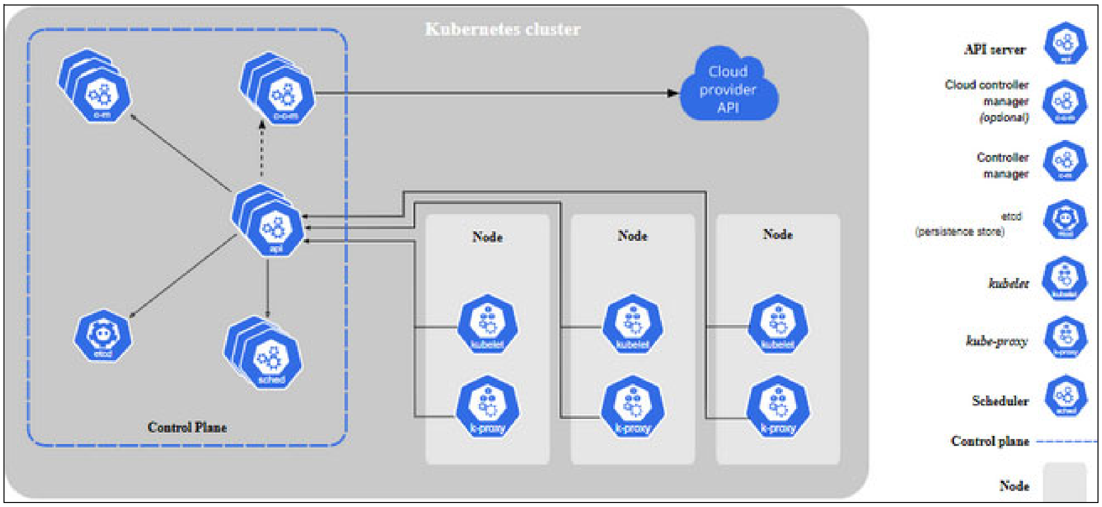


Figure 1-28 Components of a Kubernetes cluster

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 additional features and tools to enhance productivity and security, making it an ideal choice for businesses looking to leverage container technology at scale.

Red Hat OpenShift is a unified platform to build, modernize, and deploy applications at scale. Work smarter and faster with a complete set of services for bringing apps to market on your choice of infrastructure. OpenShift delivers a consistent experience across public cloud, on-premise, hybrid cloud, or edge architecture.

Red Hat OpenShift offers you a unified, flexible platform to address a variety of business needs spanning from an enterprise-ready Kubernetes orchestrator to a comprehensive cloud-native application development platform that can be self-managed or used as a fully managed cloud service.

Red Hat OpenShift provides:

- The ability to deploy and run in any environment, the flexibility to build new applications, modernize existing applications, run third-party ISV applications, or use public cloud services under a single platform.
- The tools necessary to help customers integrate data analytics, artificial intelligence and machine learning (AI/ML) capabilities into cloud-native applications to deliver more insight and value.
- Consistency and portability to deploy and manage containerized workloads, make infrastructure and investments future-ready, and deliver speed and flexibility on-premise, across cloud environments, and to the edge of the network.
- Advanced security and compliance capabilities, allowing end-to-end management and observability across the entire architecture.

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 have the option to get advanced security and compliance capability, end-to-end management and observability, and cluster data management and cloud-native data services. Red Hat Advanced Cluster Security for Kubernetes modernizes container and Kubernetes security, letting developers add security controls early in the software life cycle. Red Hat Advanced Cluster Management for Kubernetes lets you manage your entire application life cycle and deploy applications on specific clusters based on labels, and Red Hat OpenShift Data Foundation supports performance at scale for data-intensive workloads

AIX Workload Partitions

An earlier implementation closely resembling containerization was developed by IBM to run inside the AIX operating system. Introduced alongside the release of AIX 6.1 in 2007, AIX Workload Partitions (WPARs) gave AIX the ability to implement a level of operating system virtualization providing application isolation and resource control to several workloads at a time.

Preceding containerization technologies such as Docker (introduced in 2013), AIX WPARs allowed an AIX server to run several virtualized AIX environments (called a workload partitions, roughly equivalent to containers) within that single AIX image which owns all physical resources on the system (called the global environment). AIX WPARs also included the ability to run an older version of the AIX OS such as AIX 5.2 and 5.3 (called Versioned WPARs) in order to enable older applications to run on newer hardware, hence allowing customers to migrate their legacy applications to newer, more advanced IBM Power models.

AIX WPARs were also capable of being moved from one LPAR to another, or from one physical server to another, all while the application running inside the workload partition is still active (a capability called live application mobility).

Although still supported on AIX 7.3 as of December 2024, WPARs have decreased in use within AIX environments through the years. There is also no support for AIX 6.1 in a WPAR at the moment.

Note: For those interested, information on AIX Workload Partitions can be found at `https://www.ibm.com/docs/en/aix/7.3?topic=workload-partitions-aix.`

1.5.4 Automation

IBM and IBM business partners provide many Automation on IBM Power approaches from Ansible to Turbonomics.

- Ansible Collections

  IBM Power products have several Ansible collections designed to automate various tasks and streamline operations. Ansible collections are available for every management interface available on IBM Power, from the hardware level (HMC and VIOS), to the operating system level, and even for applications (SAP HANA) and databases (Oracle).Here are some of the key collections:

  - AIX Collection: Automates operations such as patching, user and group management, boot management, running commands, and managing object authority. Familiar AIX tools like NIM, alt_disk_copy and many others are available
  - IBM i Collection: Similar to the AIX collection, it includes modules for patching, user and group management, boot management, running SQL queries, and more 1.
  - Linux Collection: Provides automation for Linux environments running on IBM Power systems 2.
  - VIOS Collection: Focuses on managing Virtual I/O Servers, including creation, installation, and configuration 2.
  - HMC Collection: Includes modules for patch management, logical partition management, Power system management, password policy configurations, and dynamic inventory building 3.

  These collections help administrators efficiently manage IBM Power systems and integrate them into their automation strategies.

- IBM Turbonomics

  The monitoring and performance management of IBM Power systems can be fully automated using IBM Turbonomics. Organizations invest in IT solutions to achieve their business goals, and Turbonomics ensures they maximize their return on investment by optimizing compute, network, and storage resources at every level. Turbonomics provides a comprehensive view of resource usage, continuously monitoring and automating the most efficient allocation of resources at the lowest possible cost.

  Turbonomics can understand any application stack, whether it's in an on-premise data center, cloud, or hybrid cloud environment. By automating resource requirement decisions, Turbonomics frees up your IT engineers' time, allowing them to focus on other organizational tasks. Additionally, Turbonomics can be used to compare the costs of running existing workloads in different environments, whether on-premises or in the cloud, by better understanding the resource requirements of business applications.

Note: For more information on Turbonomics visit this IBM product website on Turbonomics.

Managing OpenShift Clusters

You can use both Red Hat Advanced Cluster Management for Kubernetes and IBM Cloud Pak for Multicloud Management to manage your OpenShift Clusters. You can also use a mix of the console and the CLI for both solutions.

Red Hat Advanced Cluster Management for Kubernetes

Red Hat Advanced Cluster Management for Kubernetes provides the capabilities to address common challenges that administrators and site reliability engineers face. Clusters and applications whether containerized or virtualized are all visible and managed from a central console, with preconfigured governance policies that can be applied consistently across environments. Users can run their operations from anywhere on Red Hat OpenShift and manage other supported Kubernetes clusters in their fleet.

Red Hat Advanced Cluster Management (RHACM) extends its multi-cluster management capabilities to environments incorporating IBM Power Systems architecture. Specifically, RHACM can manage Red Hat OpenShift clusters deployed on IBM Power, enabling centralized control and visibility alongside other Kubernetes deployments. This allows organizations to apply consistent policies, streamline application lifecycles, and monitor cluster health across diverse infrastructure, including those built on IBM Power.

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

- KVM
- VMware ESXi
- Nutanix AHV
- IBM PowerVM
- IBM z/VM®.

A full list of supported deployment options for OpenShift can be found in this documentation.

In addition, Red Hat Advanced Cluster Management supports the creation of OpenShift clusters on public clouds such as:

- AWS
- Microsoft Azure
- Google Cloud Platform (GCP)
- Microsoft Azure Government
- AWS GovCloud
- Alibaba Cloud
- Oracle Cloud Infrastructure
- IBM Cloud.

A full list of supported deployment options for OpenShift can be found in this documentation.

In essence, RHACM simplifies and automates the management of Kubernetes environments, especially those that span multiple locations or cloud providers. For more information on Red Hat Advanced Cluster Management see the RHACM data sheet.

IBM Cloud Pak for MultiCloud Management

With the IBM Cloud Pak for Multicloud Management console and CLI tools, you can view information about your clusters, add or change cluster labels, and view metering usage data for only your hub cluster.

IBM Cloud Pak for MultiCloud Management deploys its capabilities and services within a Red Hat OpenShift cluster, designated as the 'hub cluster'. This hub cluster then provides a unified view of both cloud and on-premises resources that interact with it. Specifically, IBM Cloud Pak for MultiCloud Management extends visibility to traditional workloads, including virtual machines (VMs) within virtualization clusters and standalone VMs, even those outside the OpenShift environment. This comprehensive view enhances understanding and automation of interconnected tasks across the entire infrastructure. Figure 1-29 on page 43 illustrates how IBM Cloud Pak for MultiCloud Management is deployed.

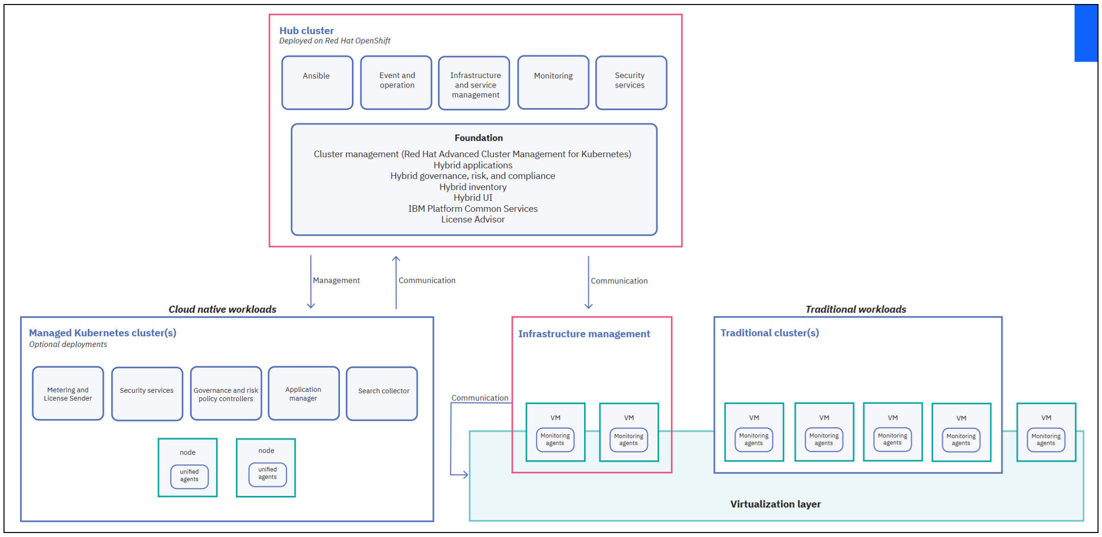
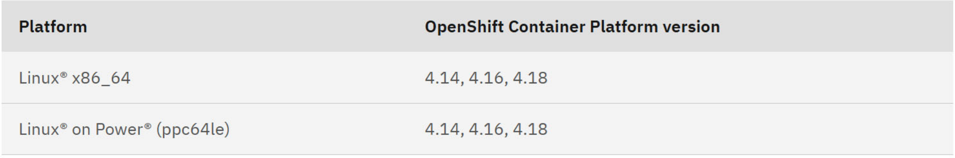


Figure 1-29 IBM Cloud Pak for MultiCloud management deployment architecture

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



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 OpenShift versions

In addition to the default features for managing multicloud environments, the IBM Cloud Pak for Multicloud Management includes the following installable modules that you can add to your cluster to manage applications and infrastructure and automate tasks:

- Monitoring Module for monitoring the performance and availability of cloud applications in hybrid cloud environments.
- Terraform and Service Automation Module for cluster security, operating efficiency, and appropriate service level delivery.
- Infrastructure Manager for controlling and managing cloud infrastructures. Red Hat Ansible Automation for running your automation tasks.

Note: For more information see Whats new in IBM Cloud Pak for Multicloud management.

1.6 IBM Power and Artificial Intelligence

The use of Artificial Intelligence (AI) is projected to unlock nearly $16 trillion in productivity by 2030[18]. Customers today expect seamless experiences and timely answers to their questions, and companies that fail to meet these experiences risk falling behind. Investment in generative AI is expected to grow 4X over the next 2 to 3 years, but it remains a small fraction of total AI spend[19], and 89% of enterprise decision makers agree that scaling AI leads to competitive differentiation[20].

[18] Fortune, April 20, 2023: IBM CEO: ‘Today’s workforce should prepare to work hand in hand with A.I.’

With AI, especially generative AI, moving from ideation to operationalization, enterprises are looking to choose the right infrastructure that is reliable, provides hybrid flexibility and trusted insights. IBM Power provides an accelerated, flexible, and safeguarded platform designed for enterprise AI workloads. Additionally, IBM Power clients have valuable data residing on their IBM Power systems, which helps them to derive trusted insights from their enterprise data and reap the benefits that AI offers.

1.6.1 Why Artificial Intelligence on IBM Power?

IBM Power provides a trusted foundation to empower your AI strategy. IBM Power systems are a good choice for AI because they offer high performance, built-in AI acceleration, and strong security features. They also integrate well with hybrid cloud environments and AI platforms like IBM watsonx®, making them versatile and efficient for AI applications.

Accelerate efficiently

AI-optimized hardware and software empower clients to accelerate AI workloads efficiently without requiring data scientists to alter their code, creating optimization directly out-of-the-box.

Improved Performance

IBM Power10 hardware comes with features optimally suited for AI workloads including an in-core accelerator called Matrix Math Accelerator (MMA). Together with the large memory capacity of IBM Power10 and high parallelism, these differentiators offer efficient and cost-effective acceleration for AI workloads. For large language models (LLMs), clients can process up to 42% more batch queries per second on IBM Power S1022 servers than compared x86 servers during peak load of 40 concurrent users and enjoy inferencing latency below 1 second.[21]

Run AI on a highly performant, sustainable platform

IBM Power10 improves the sustainability posture by providing 39% more inferencing per watt than the compared Intel-based servers.[22]

Improved Economics

Clients can leverage the parallel inferencing capabilities and higher utilization on the IBM Power platform to gain 51% lower total cost of ownership over a 3-year period running parallel inferencing in Cloud Pak for Data on IBM Power S1022 vs. compared x86 server.[23]

1.6.2 Orchestrate 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 executed.

[19] IBM Institute for Business Value, Generative AI: The state of the market
[20] Forrester Consulting Thought Leadership Paper: Overcome Obstacles To Get To AI At Scale
[21] Supporting details provided in “Improved Performance” on page 408
[22] Supporting details provided in “Run AI on a highly performant, sustainable platform” on page 408
[23] Supporting details provided in “Improved Economics” on page 408

- A combination of enterprise and/or 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 Safeguard AI and Data

Enterprise clients are concerned about safety, risk, vulnerabilities, and compliance. These are all growing areas of concern. AI models may process sensitive data at large scales and, hence, data must be safeguarded by appropriate data governance and security mechanisms.

- Simplify encryption and support end-to-end security with transparent memory encryption capabilities on IBM Power without affecting performance by using hardware features for a seamless user experience.
- Minimize latency and consolidate cryptography without having to send data to off-device accelerators with on-chip cryptographic algorithm acceleration, which allows algorithms, such as Advanced Encryption Standard (AES), SHA2, and SHA3 to run fast on IBM Power10 servers.
- Protect your applications and data with secure virtual machine (VM) isolation with orders of magnitude lower Common Vulnerability Exposures (CVEs) than hypervisors related to x86 processor-based servers.
- Security compliance profiles & real-time updates: Capabilities of PowerSC help clients centrally manage, monitor, report, and visualize security and compliance to help support compliance audits, including GDPR.

1.6.4 Hybrid flexibility

Hybrid flexibility is critical when it comes to deploying AI workloads. IBM Power provides that flexibility, enabling enterprises to harness the power of AI both on-premises and in the cloud with IBM Power Virtual Server. In addition to environment flexibility, choice matters for higher levels of the AI solution stack. IBM Power supports multiple AI-optimized software options including:

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

1.6.5 Artificial Intelligence solutions on IBM Power

Here are some solutions that enable Artificial Intelligence on IBM Power systems.

IBM Cloud Pak for Data

IBM enterprise AI solutions for IBM Power include IBM Cloud Pak for Data. IBM Cloud Pak for Data is a modular set of integrated software components for data analysis, organization and management. IBM Cloud Pak for Data on IBM Power contains a wide range of Watson, Apache, IBM Db2® and Red Hat components which help accelerate data analytics tasks within Cloud Pak for Data. As we continue to grow, additional capabilities and services within Cloud Pak for Data will be made available on IBM Power.

Open-Source Solutions

IBM Power offers community and enterprise supported open-source AI capabilities. Open-source AI solutions on IBM Power are provided through RocketCE and the Rocket AI Hub for IBM Power. RocketCE is a packaging of open-source AI tools that are optimized for IBM Power10, leveraging the IBM Power10 on-chip acceleration; available via Rocket

Software's public Anaconda channel[24]. Rocket AI Hub for IBM Power is an integrated and freely available set of best-of-breed open-source AI platform tools all optimized for IBM Power such as Katib, Kubeflow, Kubeflow Pipelines, KServe and RocketCE. All tools are delivered as container images that are operated within Kubernetes-based environments such as Red Hat OpenShift. All tools are integrated via Kubeflow and optimized to leverage unique AI hardware capabilities of the IBM Power platform.

watsonx

As the market continues to adopt foundation models for generative AI use cases, IBM Power is aligned to offer generative AI capabilities with watsonx and well positioned to deliver inferencing capabilities of foundation models. These capabilities will allow clients to deploy generative AI uses cases to improve customer experiences, increase productivity, and optimize business processes. Generative AI takes advantage of IBM Power10 on-chip acceleration to provide a differentiated experience for IBM clients. As the market continues to evolve and compute requirements change, IBM Power's mission is to provide clients with a platform that can meet the demands in a cost-effective, sustainable, resilient, and secure way.

Red Hat OpenShift

In addition to the workloads that support AI initiatives for IBM clients, advancements in Red Hat OpenShift will impact the AI solution architecture. An example of this is the introduction of OpenShift capabilities like Multi-Architecture Cluster (MAC) support. MAC enables clients to have multi-architecture OpenShift cluster with both x86 and IBM Power compute nodes. This capability allows IBM clients to deploy workloads where it makes sense, leveraging strengths and benefits of the different architectures and benefit from optimized workload deployment flexibility.

IBM Power clients now have access to a suite of AI capabilities that leverage IBM Power10's on-chip acceleration, can address the need for both enterprise and open-source solutions, and target key AI market drivers today. These capabilities allow IBM clients to tackle the most salient business challenges today by deriving actionable insights from their ever-growing multi-modal data.

[24] https://anaconda.org/rocketce/repo

2

Modernization Considerations

Modernizing applications and infrastructure is a complex undertaking, requiring careful consideration of numerous factors. As you modernize your environment, it is important to not lose focus of your key business requirements to maintain a secure and reliable infrastructure while maintain flexibility and agility to create an infrastructure that is able to quickly adapt to new challenges while creating more value from your existing data assets.

Ensure all modernization initiatives are aligned with over-arching 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 utilization. Automate processes to enhance efficiency and reduce human error. Implement comprehensive monitoring and observability solutions. Ensure compliance with all relevant regulations and industry standard.

Modernizing applications and infrastructure requires careful attention to a variety of factors, including security, storage, open-source technologies, and high availability. A well-rounded strategy that addresses these aspects can help organizations unlock the full potential of their IBM Power environment, ensuring it remains resilient, scalable, and secure in the digital age. Additionally, integrating AI and code assistants into your modernization efforts offers significant advantages by streamlining automation, boosting development productivity, maintaining high code quality, and optimizing infrastructure management. With AI-driven insights and the power of code assistants, businesses can enhance the speed, efficiency, and security of their applications, while maintaining the flexibility and scalability needed to thrive. As technology evolves, embracing these tools will be crucial for staying competitive in an increasingly fast-paced, digital landscape.

The following topics are covered in this chapter:

- 2.1, “Overview” on page 48
- 2.2, “Security” on page 49
- 2.3, “Open Source on IBM Power” on page 53
- 2.4, “Storage options” on page 55
- 2.5, “High availability and disaster recovery” on page 67
- 2.6, “AI code assistants” on page 75

© Copyright IBM Corp. 2025.

2.1 Overview

When modernizing applications and infrastructure, several critical areas should be considered to ensure a successful transition that enhances performance, security, and business continuity. Below are key factors to focus on:

1. Security

   Security should be a top priority in any modernization effort. As applications evolve and integrate with cloud platforms, APIs, and open-source technologies, ensuring robust security measures is essential. Consider the following:

   – Zero Trust Architecture (ZTA): Implement a security model that assumes no entity—inside or outside the network—is automatically trusted.

   – Data Encryption: Protect data at rest and in transit using strong encryption algorithms.

   – Identity and Access Management (IAM): Utilize multi-factor authentication (MFA) and role-based access control (RBAC) to prevent unauthorized access.

   – Regular Security Audits & Compliance: Ensure that applications meet industry compliance standards (e.g., GDPR, HIPAA, PCI-DSS) and conduct regular vulnerability assessments.

2. Use of Open Source Technologies

   Leveraging open-source software can enhance flexibility, reduce costs, and promote innovation. However, careful planning is required:

   – Containers & Kubernetes: Modernizing applications with containerization (e.g., Docker) and orchestration (e.g., Kubernetes, OpenShift) improves portability and scalability.

   – Open-Source Databases: PostgreSQL, MariaDB, and MongoDB provide cost-effective alternatives to proprietary databases.

   – Security & Maintenance: Regular patching and updates are essential for open-source software to mitigate vulnerabilities.

   – Community & Enterprise Support: Consider whether community support is sufficient or if enterprise-grade support (e.g., Red Hat, SUSE) is needed for critical applications.

3. Storage Options

   Modern applications demand flexible and scalable storage solutions to handle increasing data volumes efficiently. Key considerations include:

   – Hybrid & Multi-Cloud Storage: Using a combination of on-premises and cloud storage enables scalability and cost optimization.

   – Software-Defined Storage (SDS): Provides greater flexibility by abstracting storage resources, allowing for automation and better resource utilization.

   – NVMe and Flash Storage: These high-performance storage options can significantly reduce latency and improve application responsiveness.

   – Backup & Archival Strategies: Implement robust backup solutions, including snapshot-based backups, tiered storage, and immutable backups to protect against ransomware attacks.

4. High Availability (HA) and Disaster Recovery (DR)

   Ensuring business continuity requires planning for high availability and disaster recovery. Key strategies include:

   – Redundancy & Failover Mechanisms: Deploy HA clusters with automated failover to minimize downtime.

   - Disaster Recovery (DR) Planning: Establish a DR plan with defined Recovery Time Objectives (RTO) and Recovery Point Objectives (RPO).
   - Cloud-Based DR Solutions: Leverage cloud-based disaster recovery (IBM Cloud, AWS, Azure) for greater flexibility and faster recovery.
   - Continuous Monitoring & Testing: Regularly test HA and DR strategies to ensure they function as expected during an actual outage.
5. Use of Code Assistants

   Modern application development can be significantly enhanced with code assistants, powered by AI and machine learning. These tools improve developer productivity and code quality. Key considerations include:

   - Automated Code Generation & Refactoring: Code assistants (like IBM watsonx Code Assistant, GitHub Copilot, Tabnine, or OpenAI Codex) help developers by suggesting code snippets, completing functions, and even generating entire sections of code. These tools can speed up development cycles and reduce the potential for human error.
   - Code Review and Quality Assurance: AI-powered code assistants can also automatically check for common bugs, code smells, and inefficiencies, offering suggestions for improvements. By continuously analyzing code, they can ensure that code quality standards are consistently met.
   - Intelligent Debugging and Testing: Code assistants can help developers quickly locate and fix bugs in the code. They can also generate unit tests, automate the testing process, and identify edge cases that may have been overlooked during development.
   - Simplifying Maintenance: As applications evolve, maintaining large codebases can be challenging. Code assistants help developers understand and navigate legacy code, suggest refactoring opportunities, and ensure smooth integration with modern technologies.
   - Collaboration and Knowledge Sharing: Code assistants can enhance collaboration among development teams by ensuring consistent coding styles, practices, and standards. Additionally, they can help onboard new team members by providing context-sensitive code suggestions based on the project's existing codebase.

Modernizing applications and infrastructure requires focusing on key areas for improved performance, security, and continuity. Security should prioritize Zero Trust Architecture, data encryption, and regular audits. Open-source technologies offer flexibility and cost savings but need careful maintenance. Scalable storage solutions, like hybrid cloud and software-defined storage, are essential for managing data. High availability and disaster recovery plans, with redundancy and cloud-based solutions, ensure business continuity. AI-powered code assistants improve development efficiency, automate tasks, and ensure code quality while fostering collaboration. These areas are discussed in the following sections of this chapter.

2.2 Security

IT security is paramount in today's digital age. As businesses increasingly rely on technology to operate, protecting sensitive data and preventing cyber attacks becomes a top priority. To meet today’s security challenges, it’s essential that every layer of your company’s IT hardware and software stack remains secured. IBM Power customers utilize the most reliable mainstream server platform to innovate and get to market faster without compromising security.

IBM Power's multi-layered approach to security gives you full visibility of your hardware and software. IBM Power architecture and ecosystem provides advanced security capabilities at various levels.

At the hardware level, IBM Power servers offer advanced technology that includes tamper-resistant features built into the processor to prevent unauthorized access and modifications, secure cryptographic engines to provide strong encryption of data, and Trusted Boot to ensure that only authorized software components are loaded during system startup.

At the virtualization level, the hypervisor – which manages virtual machines – is designed to be secure and resistant to attacks. The hypervisor isolates workloads within a single physical server, allowing for secure resource sharing within your infrastructure. The Hardware Management Console (HMC) provides centralized management and control of Power systems in a secure manner.

The operating systems that run on IBM Power servers – AIX, IBM i, and Linux on Power – offer robust security features, including user authentication, access controls, and encryption support. In addition, tools such as IBM PowerSC provide a comprehensive security and compliance solution that helps manage security policies, monitor threats, and enforce compliance.

Security also requires solid management and control. Other critical elements include the implementation of data encryption for both data at rest and in flight, and strong network security processes utilizing firewalls, intrusion detection systems, and other security measures.

IBM PowerSC support

The Power10 based servers benefit from the integrated security management capabilities that are offered by:

- IBM PowerSC
- The Power software portfolio for managing security and compliance on every Power processor-based platform that is running AIX
- IBM i or the supported distributions
- Versions of Linux

PowerSC is introducing more features to help customers manage security end-to-end across the stack to stay ahead of various threats. Specifically, PowerSC 2.0 adds support for Endpoint Detection and Response (EDR), host-based intrusion detection, block listing, and Linux.

By combining these hardware, software, and management practices, IBM Power systems provide a robust foundation for security in your IT environment.

For additional information on security on IBM Power you can reference*IBM Power Security Catalog*, SG24-8568. This book describes best practices such as conducting regular security audits, keeping operating systems and applications up-to-date with the latest security patches, and implementing strong user authentication and authorization policies.

2.2.1 Encryption

When discussing encryption on IBM Power systems, it's essential to consider the breadth of the platform and the various ways data is protected. IBM Power systems employ a layered approach to encryption, addressing data at rest and in transit. IBM Power encryption protects data using a variety of methods, including hardware-based encryption, software-based encryption, and key management services.

Transparent memory encryption

The transparent memory encryption of IBM Power solutions is engineered to enable end-to-end security that meets the demanding security standards enterprises face today. It is also designed to support crypto acceleration, quantum-safe cryptography, and full homomorphic encryption to guard against future threats. Figure 2-1 shows the implementation of transparent memory encryption.

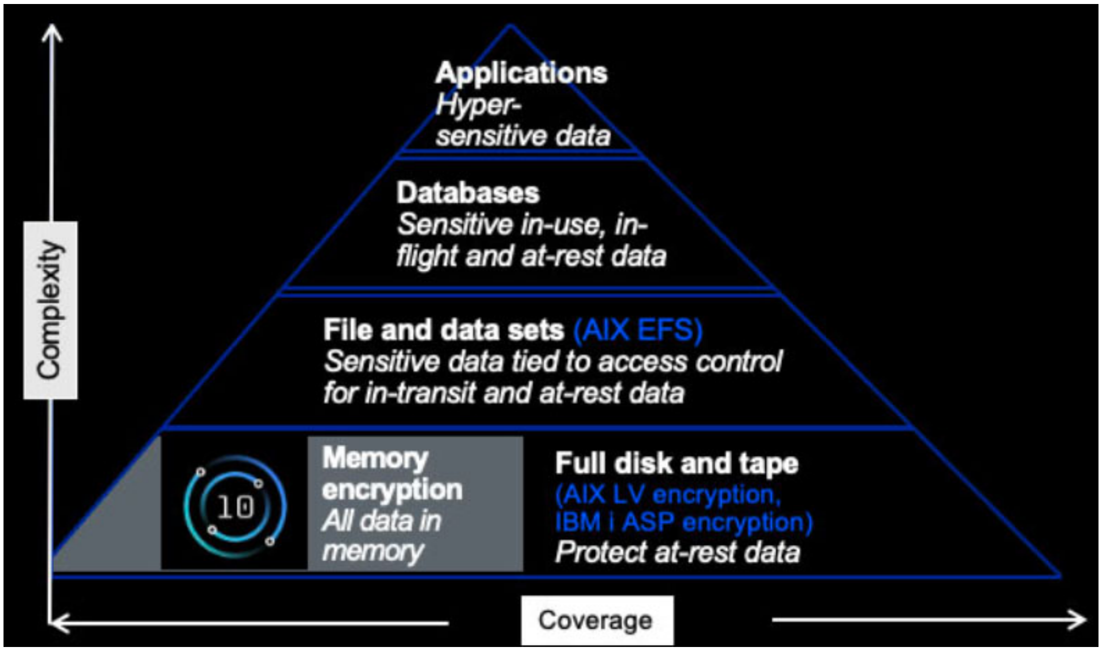

Applications
Hyper-sensitive data
Databases
Sensitive in-use, in-flight and at-rest data
File and data sets (AIX EFS)
Sensitive data tied to access control for in-transit and at-rest data
10
Memory encryption
All data in memory
Full disk and tape
(AIX LV encryption, IBM i ASP encryption)
Protect at-rest data
Complexity
Coverage

Figure 2-1 Figure - Protecting data with memory encryption.

The accelerated encryption for the newest IBM Power system model has 2.5x faster Advanced Encryption Standard (AES) crypto performance per core than that of IBM Power E980 technology[3]. Organizations can benefit from transparent memory encryption with no additional management setup.

The Power10 MCU provides the system memory interface between the on-chip SMP interconnect fabric and the OMI links. The Power10 on-chip MCU encrypts and decrypts all traffic to and from system memory that is based on the AES technology.

The Power10 processor supports the following modes of operation:

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

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

Also, memory cards that leave the data center for repair or replacement can be a potential security breach. Because the attacker might have arbitrary access to the persistent DIMM data. the stronger encryption of the AES XTS mode is required for persistent memory. The

AES XTS mode of the Power10 processor is supported for future use if persistent memory solutions become available for IBM Power servers.

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

2.2.2 Quantum Encryption

To be prepared for the Quantum era, IBM Power servers are built to efficiently support future cryptography, such as Quantum-safe cryptography and Fully Homomorphic Encryption (FHE). The software libraries for these solutions are optimized for the Power10 processor-chip instruction set architecture (ISA) and are or will be available in the respective open source communities. Future generation of IBM Power based servers will build in additional functionality to support Quantum-safe computing.

Quantum-safe encryption

Quantum-safe encryption (QSE), also known as Post-Quantum Cryptography (PQC), refers to encryption methods that are secure against both classical and quantum computers. As quantum computers advance, they might pose a threat to existing cryptographic systems, potentially compromising their security. QSE is essential for protecting sensitive data, communication channels, and user identities in the age of quantum computing.

The urgency of adopting QSE stems from two primary concerns: Advanced quantum computers might enable adversaries to intercept and decrypt protected digital communications through Harvest Now, Decrypt Later (HNDL) strategies, even before reaching Q-Day. (Q-Day is the anticipated point in time when quantum supremacy becomes widespread and many of the current encryption algorithms are no longer effective.) Migrating to QSE might require over a decade due to the complexities of organizational structures and IT infrastructure.

Therefore, organizations should start evaluating and implementing QSE solutions immediately to ensure continued protection and maintain trust among their stakeholders. Delaying QSE adoption might have severe consequences. Legacy cryptographic systems left unaltered might be compromised if there is a successful quantum attack, which can expose sensitive data and risk confidential business transactions and individual privacy. Financial institutions, critical infrastructure providers, and government agencies might face significant challenges in maintaining operational integrity and confidentiality.

Prioritizing QSE implementation is crucial for long-term cybersecurity resilience. Power10 supports these quantum-safe algorithms to help ensure robust security even as quantum computing advances.

Fully Homomorphic Encryption

Fully Homomorphic Encryption enables computations to be performed directly on encrypted data without decrypting it first to help ensure that sensitive data remains confidential even during processing. FHE operates at the software level and involves sophisticated mathematical algorithms to enable computations on ciphertexts. Implementing FHE requires specialized libraries and frameworks The software libraries for these solutions are optimized for the Power processor-chip ISA. However, FHE is computationally intensive and can introduce performance overhead compared to conventional hardware-only encryption methods due to the complexity of the algorithms.

2.3 Open Source on IBM Power

Open source software (OSS) is source code 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 as they see fit – typically under a specific license. Since the source code is “freely” available, there is usually no cost to use OSS. However, in terms of OSS, the term “free” refers to freedom and not price. Freedom to execute code, freedom to study source code, freedom to redistribute, and freedom to improve.

Some key points about open source software (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

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

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

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

  – Linux: A widely used operating system
  – Mozilla Firefox, a web browser originally based on Netscape Navigator
  – Apache HTTP Server: A web server
  – LibreOffice, a suite of office productivity apps that rival Microsoft Office

Usually, the source code for OSS is made available on a code sharing platform such as GitHub (see section 6.4.1, “Github” on page 190 for more information). Additionally, 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 as well as IBM z and LinuxOne. For more detail see https://osuosl.org/.

Support for Open Source Software

When considering the use of open source software for your business applications, you need to understand how the OSS you are planning on using is supported. It is crucial to understand that many OSS projects are offered “as-is.” This means the source code is available, but dedicated vendor support isn't included. Users encountering issues, bugs, or needing new features are typically expected to consult documentation and leverage the community-driven development process.

Some OSS projects do offer enterprise support – generally for a fee. For example Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES)) both provide enterprise support for their distributions, but this is not the norm. Fedora and openSUSE are also OSS Linux distributions, but they lack formal enterprise support. For some popular open source projects you can buy a support contract.

Open Source at IBM

IBM partners with most of the major open source communities that drive today's businesses. Our developers are collaborators and committers, encouraging open governance, contributing code, helping with licensing, and pushing the technology forward. Open source is alive and well in IBM, with thousands of IBMers participating in projects to expand technologies and strengthen communities. At our core, we believe that open source is the bedrock of modern computing.

From our work with Linux, Apache, and Eclipse in the early years of open source to our current work across all aspects of the cloud native ecosystem, artificial intelligence, quantum computing, and machine learning, IBM has demonstrated 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 our clients.

Open Source on Power

IBM Power servers are built with processors that support both big-endian and little-endian operating systems which are supported by AIX, IBM i and Linux (ppc64le). IBM has built in support for open source software on both AIX and IBM i. As Linux is built on an open source foundation, open source software is supported by default.

To run open source software on AIX, you primarily use the "AIX Toolbox for Open Source Software," which provides a collection of pre-built open source packages specifically designed for the AIX operating system, allowing you to install and manage them using tools like DNF (the recommended package manager) to access a wide range of popular open source applications. For more information see AIX Toolbox overview.

Similar to AIX, IBM i supports many open source products. Information on how to install open source packages can be found at `http://ibm.biz/ibmi-rpms.` As part of IBM i software maintenance agreement (SWMA), IBM i has an active open source community. See the community links on the IBM i Open Source Resources page to join. See the IBM i open source documentation for general documentation regarding open source, as well as the other documentation links on the IBM i Open Source Resources page.

IBM supports the installation of IBM-delivered RPM packages. on IBM i. This support includes software download assistance or surrounding tools such as Access Client Solutions. This does not include actual usage or defect support of these packages.

When considering open source applications for IBM Power hardware check out the project's Github page. Read the documentation carefully to understand if the project supports different CPU architectures such as ppc64le. It is helpful to read through Github Issues and Pull Requests (PRs) to see if there has been a discussion about supporting IBM Power hardware. Additionally, you can read through the Github Releases page to see if the project provides pre-built binaries or packages for IBM Power hardware.

One very helpful tool to find open source software is the IBM Open Source POWER® availability tool.

2.4 Storage options

The IBM Power systems can be attached with all storage types. The type attached is determined by the purpose and objectives of the solution.

The following sections describe the various storage types understood to exist in a modern technology environment, and then goes on to describe some suggestions for provisioning storage with IBM Power systems.

2.4.1 File Storage

File systems organize data in a hierarchical structure of directories and files, making them intuitive and user-friendly for everyday tasks. Initially, they were designed for direct-attached storage devices dedicated to individual servers. However, support for remotely attached and shared storage is now provided through Network Attached Storage (NAS) devices, which can be provisioned to one or more servers. These NAS devices connect via protocols such as Network File System (NFS) or Common Internet File System (CIFS) over Ethernet. Input/output (I/O) access to the file system is achieved by reading or writing individual files from the shared namespace, allowing multiple users to share and access the data simultaneously.

NAS storage solutions work very well for smaller environments, however as the number of servers access the storage scales, the performance of the storage solution can suffer. IBM has two highly scalable solutions available for file storage, IBM Spectrum® Scale and IBM Ceph.

IBM Storage Scale, formerly known as GPFS (General Parallel File System), is a high-performance clustered file system designed for large-scale storage environments. It can be deployed in shared-disk or shared-nothing distributed parallel modes, or a combination of these. Storage Scale is known for its ability to provide concurrent high-speed file access to applications running on multiple nodes within a cluster. It supports various features such as Information Lifecycle Management (ILM), Active File Management (AFM), and Clustered NFS (CNFS). Spectrum Scale is used in many of the world's largest commercial companies and supercomputers, including the Summit supercomputer at Oak Ridge National Laboratory. IBM Storage Scale is discussed further in section 2.4.6, “IBM Storage Scale” on page 59.

IBM Ceph is an enterprise-class software-defined storage solution designed for data-intensive applications. It is built on the open-source Ceph storage system and provides a scalable, reliable, and flexible platform for managing block, file, and object storage. IBM Ceph is particularly suited for cloud infrastructure and web-scale object storage, offering features like data replication, erasure coding, and self-healing capabilities. It supports various interfaces, including RESTful APIs (S3/Swift), block device interfaces, and filesystem interfaces. IBM Ceph is designed to handle vast amounts of data, making it ideal for modern data storage needs. Additional information on IBM Ceph can be found in *IBM Storage Ceph Concepts and Architecture Guide*, REDP-5721.

For more information on file system solutions see:

– https://www.ibm.com/products/storage-scale
– https://www.ibm.com/docs/en/storage-ceph/7.0.0?topic=introduction-storage-ceph

2.4.2 Block Storage

Block storage, or block-level storage, organizes data into uniform blocks for efficient storage on SANs or cloud platforms. This method divides data, such as files or database entries, into equally sized blocks, optimizing their placement on physical storage for rapid access and retrieval. Data interaction occurs by reading or writing individual blocks, allowing the system to strategically distribute data for optimal performance. This approach excels in high-transaction environments with large databases, prioritizing speed.

Block storage frequently employs logical volumes provisioned to servers within a SAN, utilizing protocols like SCSI or Fibre Channel. Modern block storage solutions increasingly leverage solid-state drives (SSDs) over traditional spinning disks, and the NVMe protocol further enhances performance with increased throughput and reduced latency, catering to high-performance enterprise needs.

> **Note:** For an introduction on block storage please read this article.
> https://www.ibm.com/think/topics/block-storage

2.4.3 Object Storage

In an object storage system, data is stored in a flat namespace that scales to trillions of objects and is optimized to store unstructured data, such as documents, images, audio, and video files. Object storage simplifies how users access data, supporting new types of applications and allowing users to access data by various methods, including mobile devices and web applications.

https://www.ibm.com/docs/en/storage-insights?topic=systems-object-storage

The vast majority of cloud storage available in the market leverages an object-storage architecture. Some notable examples are;

- Amazon Web Services S3, which debuted in March 2006,
- Microsoft Azure Blob Storage,
- Rackspace Cloud Files (whose code was donated in 2010 to OpenStack project and released as OpenStack Swift),
- Google Cloud Storage released in May 2010.

https://en.wikipedia.org/wiki/Object_storage

OpenStack Swift is an open source object storage system that is widely used for cloud storage.

> **Note:** For an introduction on Object Storage please visit
> https://www.ibm.com/think/topics/object-storage

2.4.4 Container Storage Terminology and Concepts

Kubernetes offers various types of volumes to provide storage for containers. Each volume type is designed for specific use cases within a containerized environment. A container can utilize multiple volume types at the same time. This section explains the different volume types that can be used by a container.

Ephemeral Volumes

During a containers runtime the storage used is ephemeral, self-contained and portable but is not tied to persistent storage unless defined in the container’s StorageClass property. An ephemeral read/write layer is created that handles all written data and is not persisted. When the container stops, whether intentionally or unintentionally terminated, all the underlying read/write data layer disappears along with the container.

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

Kubernetes supports several different kinds of ephemeral volumes for different purposes:

- emptyDir: empty at Pod startup, with storage coming locally from the Kubelet base directory (usually the root disk) or RAM
- configMap, downwardAPI, secret: inject different kinds of Kubernetes data into a Pod
- image: allows mounting container image files or artifacts, directly to a Pod.
- CSI ephemeral volumes: similar to the previous volume kinds, but provided by special CSI drivers which specifically support this feature
- generic ephemeral volumes, which can be provided by all storage drivers that also support persistent volumes
- emptyDir, configMap, downwardAPI, secret are provided as local ephemeral storage. They are managed by Kubelet on each node.

For more information on ephemeral storage see
`https://kubernetes.io/docs/concepts/storage/ephemeral-volumes`/

Persistent Volume

A persistent volume is a low level representation of a storage volume that is assigned to a container. As the name implies, the data on a persistent volume remains available even after the container is removed and can be accessed when the container is re-initialized. This persistence is accomplished utilizing Persistent Volume Claims (see “Persistent Volume Claims” on page 58) built into the container definition.

PVs are defined by an OpenShift PersistentVolume API object, which represents a piece of existing storage in the cluster that was either statically provisioned by the cluster administrator or dynamically provisioned using a StorageClass object.

PVs are volume plugins like Volumes but have a lifecycle that is independent of any individual pod that uses the PV. PV objects capture the details of the implementation of the storage, be that NFS, iSCSI, or a cloud-provider-specific storage system. A PV is anything that can be accessed by the hardware and underlying operating system.

A persistent volume can be provisioned in the following access modes to a container environment.

- ReadWriteOnce (RWO) — volume can be mounted as read-write by a single node.
- ReadOnlyMany (ROX) — volume can be mounted read-only by many nodes.
- ReadWriteMany (RWX) — volume can be mounted as read-write by many nodes.
- ReadWriteOncePod (RWOP) — volume can be mounted as read-write by a single Pod.

OpenShift Container Platform supports the following persistent volume plugins:

- AWS Elastic Block Store (EBS)
- AWS Elastic File Store (EFS)
- Azure Disk
- Azure File
- Cinder
- Fibre Channel
- 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 persistent volumes (PV) representing locally-mounted file systems. This is the lowest level of physical volume attachment for a worker node, and can be seen as simply attaching SAN block storage or making LUNs available for storage to a worker node or VM.

In previous versions of OpenShift, the attachment of these volumes required some additional manual tasks to be visible and utilized by OpenShift, but now administrators can use the OpenShift Local Volume Operator.

Persistent Volume Claims

A persistent volume claim (PVC) is specific to an OpenShift project, and is created and used by developers as a means to access a PV.

PV resources are not attached to any single project, and they can be shared across the entire OpenShift Container Platform cluster and claimed from any project. After a PV is bound to a PVC, that PV can not then be bound to additional PVCs. This has the effect of scoping a bound PV to a single namespace, that of the binding project. In general.

- A PVC is a binding between a pod and PV.
- The PVC is a request for a PV type and access mode.
- The PVC is bound to the same namespace as the pod.
- Kubernetes looks for a PV that meets the criteria defined by the requesting Pod, and if there is one, it matches the claim to the PV.
- Claims can request specific size and access modes (e.g., they can be mounted ReadWriteOnce, ReadOnlyMany or ReadWriteMany).

Note: For more information on OpenShift Storage please visit this partner page.

Storage Class

A Storage Class allows dynamic provisioning of Persistent Volumes, when a PVC claim is made by a Pod or a yaml definition. A StorageClass abstracts the underlying storage provider.

A StorageClass has a backend provisioner that determines what volume plugin is used for provisioning PVs. The dynamics of the storage class is provided by the Container Storage Interface Drivers (CSI) that are specific to the storage platform or cloud provider to give Kubernetes access to the physical storage.

Each storage backend has it's own provisioner. Storage Backend is defined in the StorageClass component *via provisioner attribute*. The autonomy that CSI brings allows greater response, scalability, and management of the platform as a whole, including better use of the underlying infrastructure

2.4.5 IBM Block Storage CSI Driver

The CSI was designed with the objectives of being an open specification for exposing block and file storage systems to container orchestration systems, Kubernetes and OpenShift.

IBM features the following written CSI driver families:

- IBM block storage CSI driver, which is used by Kubernetes for persistent volumes, 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 Spectrum Scale CSI driver, for file-based storage.

Note: For additional information on the IBM Block Storage CSI Driver reference the Redbook publication *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 known as IBM Spectrum Scale) is a high-performance, scalable storage solution designed to manage and store large amounts of data across distributed environments. It provides a unified storage platform that supports file, object, and block storage. IBM Storage Scale is designed for enterprises with demanding workloads, such as artificial intelligence (AI), analytics, and high-performance computing.

IBM Storage Scale is IBM's strategic high-performance parallel file system, a shared storage platform for end-to-end collaborative common enterprise, data platform, big data analytics, and AI workflows (see Figure 2-2).

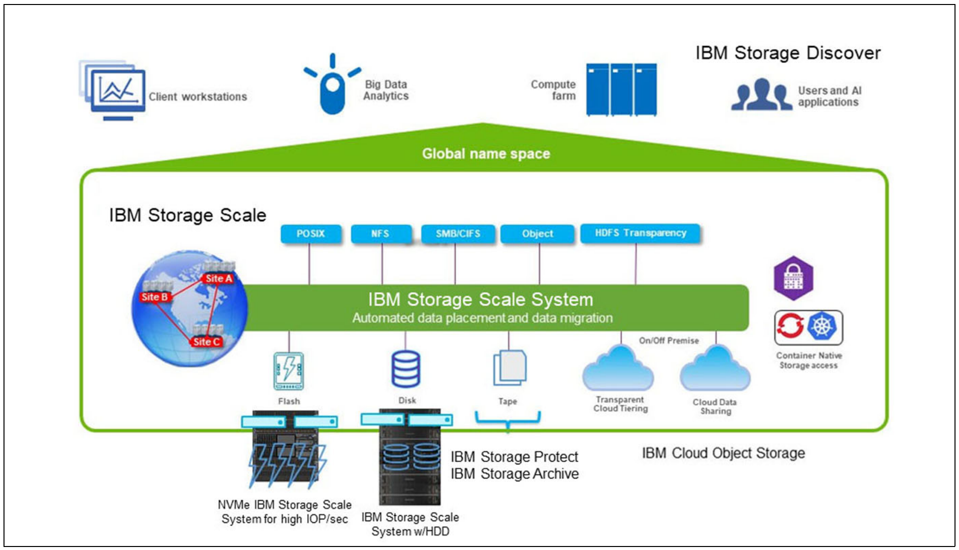


Figure 2-2 IBM Storage Scale System overview

IBM Storage Scale is designed to provide the following major value propositions:

- Simplified data management by supporting enterprise workflows on a single common enterprise data platform.
- A single global namespace that supports enterprise-level data over high-performance networks.
- Enables intelligent automatic tiering of data between storage pools, externally to tape, to object based and cloud resources. This delivers cost-effective storage economics by automatically managing and tiering data to different classes of storage.

Figure 2-3 shows some of the solutions that utilize IBM Storage Scale.

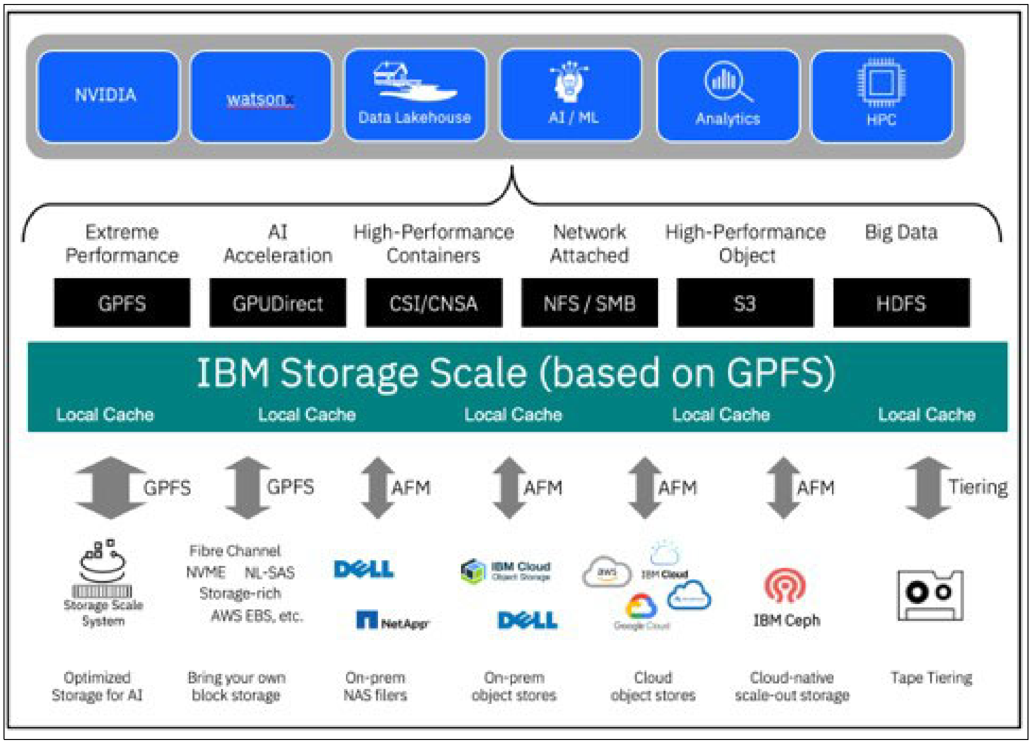


Figure 2-3 IBM Storage Scale solutions

IBM Storage Scale software provides organizations with a global data platform optimized for today's most demanding unstructured data workloads

Storage Scale is based on a massively parallel file system and can be deployed on multiple hardware platforms including x86, IBM Power, IBM z mainframes, ARM-based POSIX client, virtual machines, and Kubernetes.

- IBM Storage Scale is a software-defined file and object storage for both structured and unstructured data;
- IBM Storage Scale System 6000 is a hardware implementation of Storage Scale software and is optimized for the most demanding AI, HPC, analytics, and hybrid cloud workloads. The IBM Storage Scale System 6000 is shown in Figure 2-4 on page 61. The IBM Storage Scale System 6000 can deliver up to 310GB/s throughput.
- IBM Storage Scale System 3500 is for customers requiring an enterprise-ready entry-level or mid-level system.

Figure 2-4 IBM Storage Scale System 6000

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

IBM Storage Scale users are unaware of the physical distribution of data in the IBM Storage Scale data server physical storage pools. The automatically balanced data distribution is seamlessly determined by the IBM Storage Scale policy engine at the time that the data is imported. The policy engine can also transparently move data from one storage pool to another storage pool while the data is accessed and active.

The IBM Storage Scale parallel file system provides an enterprise the capability for data management over large amounts of data, while also performing constant auto-balance of workload and storage by equally distributing I/O and data within a storage pool or among different storage pools.

The preferred method of accessing IBM Storage Scale data is to install the IBM Storage Scale client on every workstation or server that accesses IBM Storage Scale data. The IBM Storage Scale client provides the multiple threads and communication with multiple data servers to provide high-performance parallel throughput. While doing so, IBM Storage Scale also manages full read/write data integrity between multiple users who are working with the data in the file system.

IBM Storage Scale for NVIDIA

IBM Storage Scale is the preferred storage system for NVIDIA solutions. IBM Storage Scale System is an NVIDIA certified ultra-performance solution that drives 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, IBM Storage Scale System delivers six nines of data reliability, cyber resiliency, and multi-protocol data pipelines for the most demanding enterprises. Software-defined IBM Storage integrates and tiers your data, so you can leverage a global data platform to bring value to your organization and transform data-intensive AI workloads into actionable insights. This is shown in Figure 2-5 on page 62.

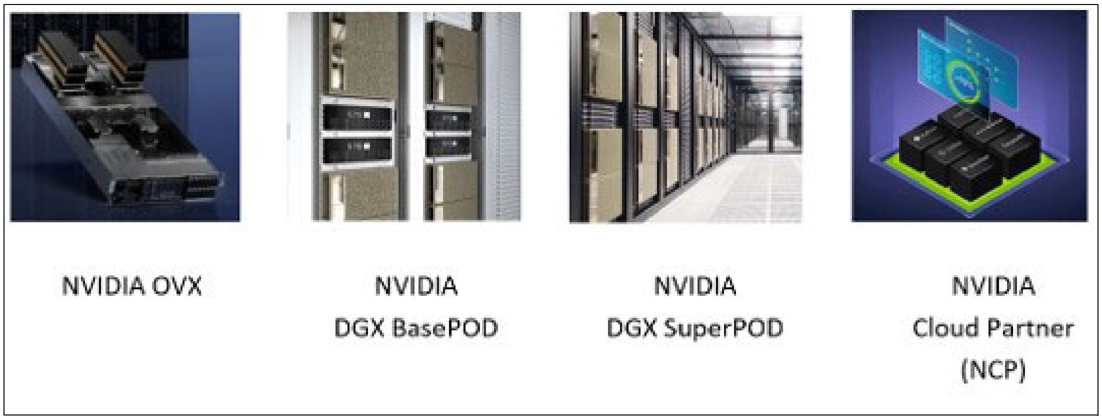


Figure 2-5 NVIDIA systems solutions with IBM Storage Scale

The unstructured and semi-structured data from AI workloads, advanced analytics, data lakes, and other data-intensive apps must be stored in distributed file and object systems to make it accessible to geographically dispersed applications, services, and devices.

IBM Storage Scale software is designed to address these requirements with global data abstraction services that provide connectivity from multiple data sources and multiple locations to bring together data wherever it lives, including non-IBM storage environments.

2.4.7 IBM Storage Fusion

IBM Storage Fusion is a container-native hybrid cloud data platform that offers simplified deployment and data management for Kubernetes applications on Red Hat OpenShift Container Platform. IBM Storage Fusion is designed to meet the storage requirements of modern, stateful Kubernetes applications and to make it easy to deploy and manage container-native applications and their data on Red Hat OpenShift Container Platform. IBM Fusion it is an advanced storage and backup solution that is designed to simplify data accessibility and availability across hybrid clouds. Companies can expand data availability across complex hybrid clouds for greater business performance and resilience. With the IBM Storage Fusion solutions, organizations manage only a single copy of data. They need not create duplicate data when moving application workloads across the enterprise, easing management functions when you streamline analytics and AI.

A Red Hat OpenShift cluster administrator must properly configure storage before installation of any IBM Cloud Pak Product, and should understand any limitations that are associated with the storage you plan to use. Not all services support all types of storage. You could connect and configure a mixture of different storage providers to satisfy your organizations needs. However, an administratively efficient solution is to use one storage provider for all your storage requirements.

IBM Cloud Pak Products need high performance storage solutions, and use Container Storage Interface (CSI) drivers from platform-providers like IBM Storage Scale, AWS EFS, and Azure Files. Be careful about using open source storage providers that might have limitations such as a limit on the number of connections or other limitations. IBM Storage Fusion not only provides the high performance Storage Scale software-defined storage management software, but provides data protection in consolidating all IBM Storage Protect solutions. providing a comprehensive solution to storage and data protection for all container workloads. This is shown in Figure 2-6 on page 63.

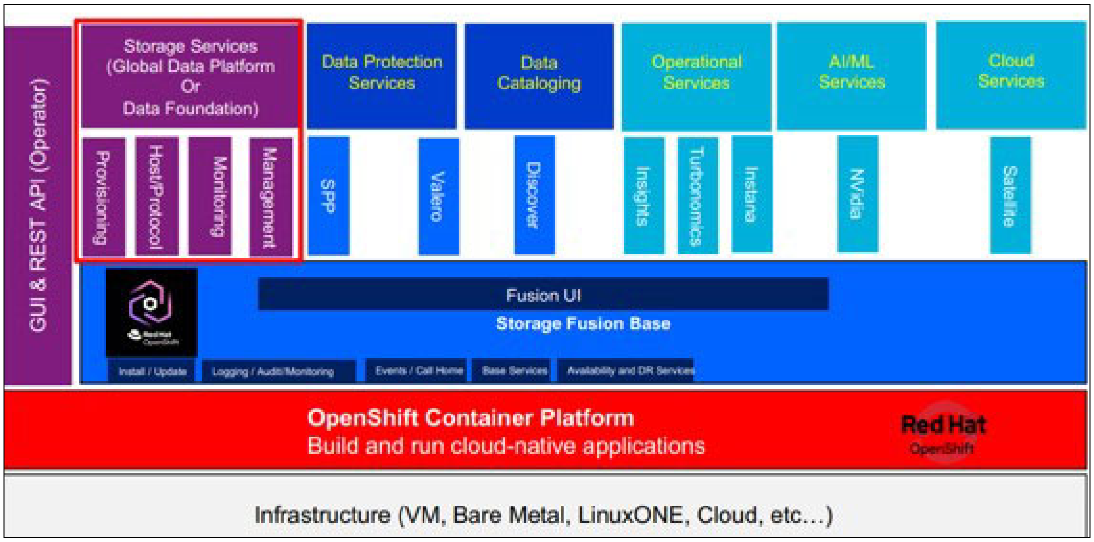


Figure 2-6 Storage Fusion General Architecture

Fusion Offerings

There are two offerings for Storage Fusion.

- As the software deployment on existing user provisioned hardware.
- As an appliance with the Hyper-converged Infrastructure (HCI) appliance.

IBM Fusion

IBM Fusion is a software-defined storage management software with protection, backup, and caching elements and can be run on existing hardware resources. IBM Fusion is supported on a number of platforms and clouds.

IBM Fusion HCI System

IBM Fusion HCI is a purpose-built, hyper-converged architecture that is designed to deploy bare metal Red Hat OpenShift container management and deployment software alongside IBM Fusion software. For consistent and rapid deployment and management, it features an appliance form-factor, hyper-converged infrastructure along with integrated software-defined storage to meet the storage requirements of modern, stateful Kubernetes applications. Built with a storage platform that includes the essential elements necessary for mission-critical containers and hybrid cloud, the IBM Fusion provides a comprehensive infrastructure with compute, networking, and storage resources, including a data platform and global data services for Red Hat OpenShift.

> **Note:** To learn more about IBM Fusion HCI System, see IBM Fusion documentation.

The storage component of Fusion also has two different offerings.

- Global Data Platform - Containerized IBM Storage Scale
  - Storage efficiency with Storage Scale RAID
  - Active File Management for geographic distance extension
  - Scale out parallel file system

  Potential use cases for this option are:

  - Watsonx
  - Db2 Warehouse
  - Metro D/R or Regional D/R

- Data Foundation - Ceph along with other containerized services
  - Block, file and object storage (CephRBD, CephFS, RGW)
  - Consistent architecture and storage classes across multiple infrastructures
  - Simple lifecycle management

  Potential use cases for this option are other basic storage provisioning requirements.

Storage and Performance Validation

In order to determine if your storage attached to your Kubernetes or OpenShift cluster is compatible with IBM Cloud Paks, and able to provide the required I/O performance to supports the applications, you can run validation tools. IBM provides a storage validation tool and a performance validation tool.

Storage validation tool

Run the IBM Software Hub storage validation tool on your Red Hat OpenShift cluster to verify your storage setup for use with IBM Software Hub. Set up your environment and before you install your IBM Cloud Paks, run the validation tool. For more information about the tool, see storage validation tool.

Performance validation tool

Run the storage performance tool to ensure that your storage performs properly and you meet the recommended performance guidelines. Collect Storage performance metrics on your Red Hat OpenShift cluster. For more information about the tool, see k8s-storage-perf GitHub repository.

What's new in IBM Storage Fusion 2.9

IBM Fusion 2.9.0 includes new features in the following areas:

- New platform support. You can now deploy IBM Fusion on Amazon Web Services ROSA HCP platform.
- Parallel upgrades. Support is available for parallel upgrades for IBM Fusion operator components that are not part of rolling updates. This enhancement allows multiple components to be upgraded simultaneously, significantly reducing downtime and improving overall system efficiency.
- Automatic Backup & Restore service upgrade based on the Backup & Restore service upgrade availability.
- Improvements to Multi-cluster IBM Fusion using Hosted Control Plane. Enhanced ability to deploy IBM Fusion services from the Red Hat Advanced Cluster Management for Kubernetes.
- Simplified image mirroring during installation and upgrade
- Backup & Restore enhancements.
- Change Block Detection Support. Backup & Restore now has change block detection for Ceph RBD block volumes. With this change, the DataMover does not have to process the entire volume to identify changes. This feature greatly reduces backup times for applications using Ceph RBD block mode volumes, including OpenShift Virtualization VMs based on RBD block volumes.
- Self-service Backup & Restore. You can protect your namespace application with IBM Fusion Backup & Restore even as an application user without a cluster or a user without IBM Fusion administration rights.

For further details on what's in Storage Fusion please see the product documentation.

2.4.8 Fusion Deployment Options

As discussed earlier, Fusion is a software defined storage (SDS) solution running on Red Hat OpenShift. With the Fusion HCI product, IBM provides a hyperconverged infrastructure implementation installed on a flexible hardware platform that can be rolled into your infrastructure and implemented in a matter of hours.

The Fusion SDS product supports environments running on Red Hat OpenShift either on premise or in cloud providers. Figure 2-7 shows environments where Fusion can be deployed.

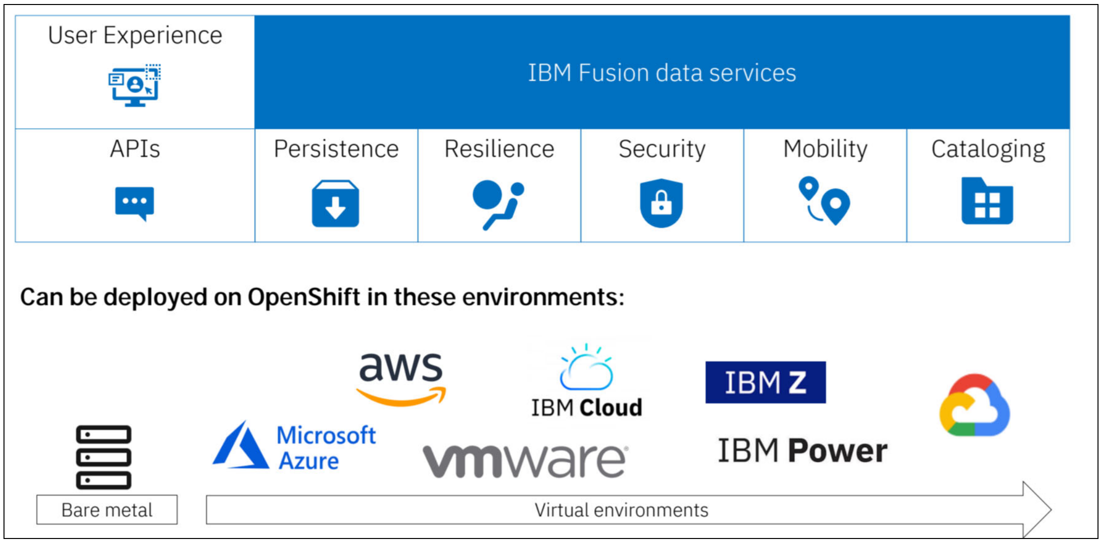
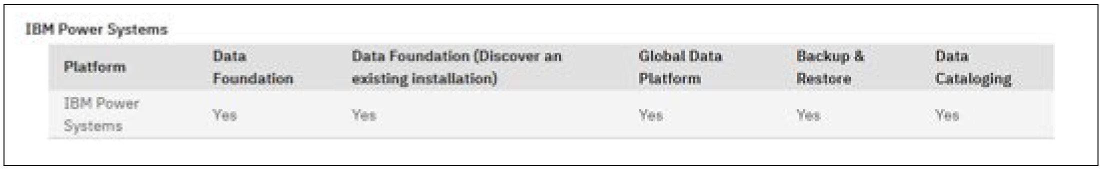


Figure 2-7 Fusion deployment options

For full list of supported services by platform reference this support matrix. In the next section we will focus on Fusion running on IBM Power.

IBM Fusion on Premises IBM Power

IBM Fusion on IBM Power provides all the capabilities of 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 Fusion support for IBM Power

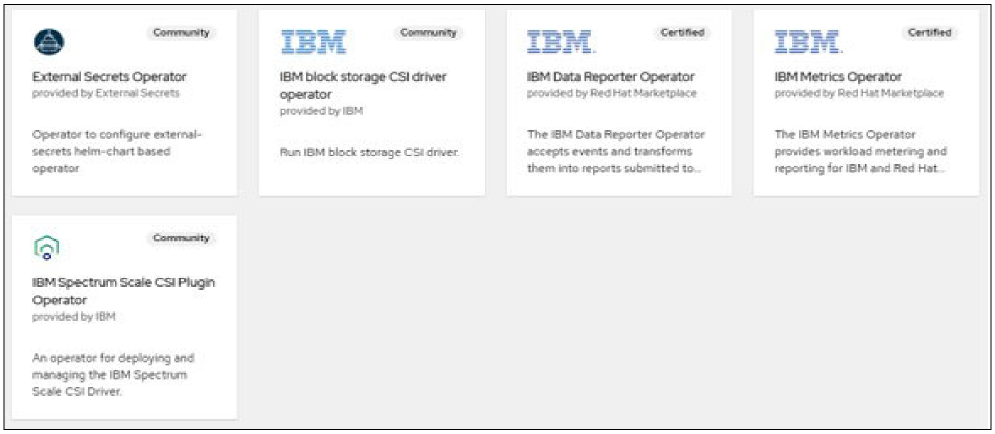
Installation considerations

The following considerations need to be addressed:

- You should have a Red Hat OpenShift cluster installed prior to installing IBM Fusion on IBM Power. IBM Fusion is provided as an operator in the Red Hat OpenShift Container Platform web management console.
- Before installing, you need to have three additional worker nodes with at least the minimum resource requirements in CPU, memory and attached SAN storage for the Fusion discovery process to find the storage workers. For system requirements see Fusion System Requirements. This lists the CPU, memory and storage requirements for

the cluster based on the Fusion components being configured. The selection and configuration of these workers nodes is performed from the Storage Fusion GUI interface.

- When selecting storage, SSD or NVMe devices are the recommended option for optimal performance and reliability. HDDs are supported for development or test only. For more information on configuring storage refer to IBM Fusion Storage Configuration.
- When a new OpenShift cluster is installed on IBM Power you will be provided with some free operators for manual configuration of storage provisioning. These include the CSI drivers for IBM Block Storage and IBM Spectrum Scale as shown in Figure 2-9.



Community
External Secrets Operator
provided by External Secrets
Operator to configure external-secrets helm-chart based operator

Community
IBM block storage CSI driver operator
provided by IBM
Run IBM block storage CSI driver.

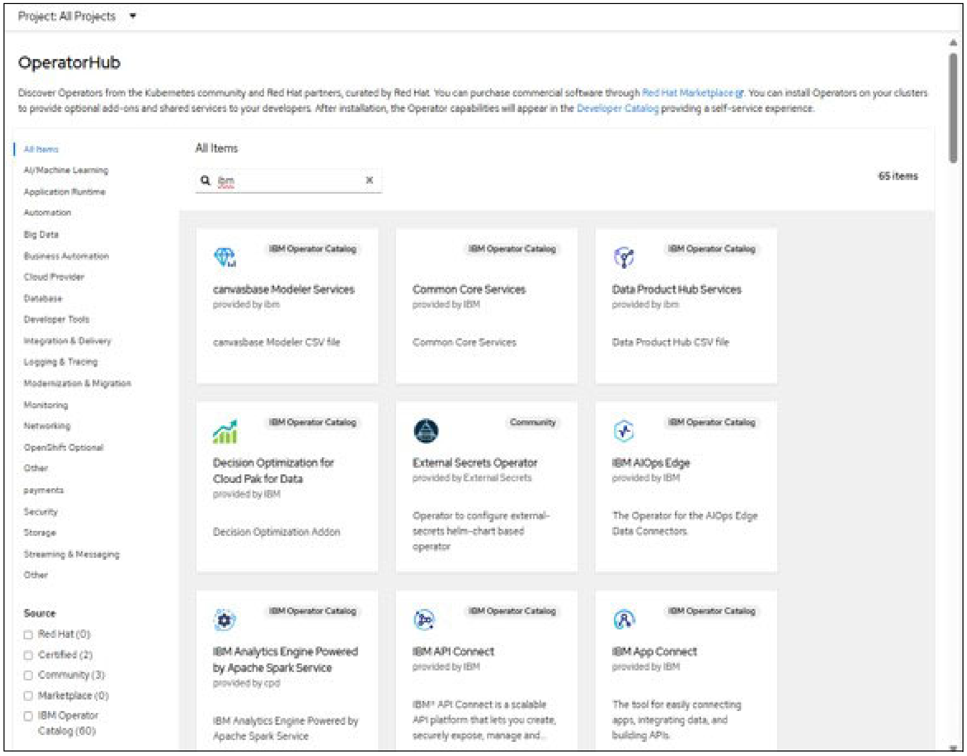
Certified
IBM Data Reporter Operator
provided by Red Hat Marketplace
The IBM Data Reporter Operator accepts events and transforms them into reports submitted to...

Certified
IBM Metrics Operator
provided by Red Hat Marketplace
The IBM Metrics Operator provides workload metering and reporting for IBM and Red Hat...

Community
IBM Spectrum Scale CSI Plugin Operator
provided by IBM
An operator for deploying and managing the IBM Spectrum Scale CSI Driver.

Figure 2-9   Operator catalog default view

However, when you enable the IBM Operator catalog, you can see the full suite of IBM products including Fusion is available for installation as shown in Figure 2-10.



Project: All Projects

OperatorHub

Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. You can purchase commercial software through Red Hat Marketplace. You can install Operators on your clusters to provide optional add-ons and shared services to your developers. After installation, the Operator capabilities will appear in the Developer Catalog providing a self-service experience.

All Items
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
payments
Security
Storage
Streaming & Messaging
Other

Source
□ Red Hat (0)
□ Certified (2)
□ Community (3)
□ Marketplace (0)
□ IBM Operator Catalog (60)

All Items

ibm

65 items

IBM Operator Catalog
canvasbase Modeler Services
provided by ibm
canvasbase Modeler CSV file

IBM Operator Catalog
Common Core Services
provided by IBM
Common Core Services

IBM Operator Catalog
Data Product Hub Services
provided by ibm
Data Product Hub CSV file

IBM Operator Catalog
Decision Optimization for Cloud Pak for Data
provided by IBM
Decision Optimization Addon

Community
External Secrets Operator
provided by External Secrets
Operator to configure external-secrets helm-chart based operator

IBM Operator Catalog
IBM AIOps Edge
provided by IBM
The Operator for the AIOps Edge Data Connectors

IBM Operator Catalog
IBM Analytics Engine Powered by Apache Spark Service
provided by cpd
IBM Analytics Engine Powered by Apache Spark Service

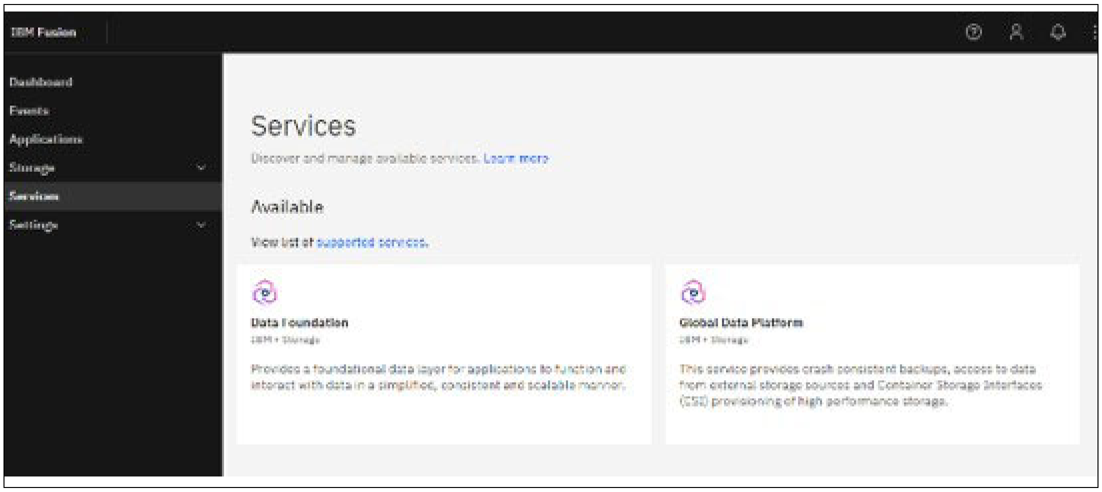
IBM Operator Catalog
IBM API Connect
provided by IBM
IBM® API Connect is a scalable API platform that lets you create, securely expose, manage and...

IBM Operator Catalog
IBM App Connect
provided by IBM
The tool for easily connecting apps, integrating data, and building APIs.

Figure 2-10   View of IBM operators

- The Fusion operator will not create a storage cluster unless a valid Fusion license is attached to the administrators IBM Container registry key. You can provide your license key, or you can obtain a Trial Fusion license from this location. Once you have the key, then configure the Fusion pull secret in Red Hat OpenShift, and you are ready to install Fusion from the operator. Instructions for creating the pull secret are found here.

Once installed, you can select to create the Data Foundation option for the ODF Ceph Cluster or choose Global data platform which utilizes IBM Storage Scale as shown in Figure 2-11.



Services

Discover and manage available services. Learn more

Available

View list of supported services.

Data Foundation

IBM • Storage

Provides a foundational data layer for applications to function and interact with data in a simplified, consistent and scalable manner.

Global Data Platform

IBM • Storage

This service provides crash consistent backups, access to data from external storage sources and Container Storage Interfaces (CSI) provisioning of high performance storage.

Figure 2-11 Storage configuration choice in Fusion

For full installation documentation see Installing IBM Fusion on premises on IBM Power.

2.5 High availability and disaster recovery

In an age where digital technologies underpin every facet of our lives, from critical business processes to personal communications, one aspect stands as paramount: uninterrupted access to data, applications, and services. The modern world demands nothing less than a seamless, always-on experience, and any interruption (be it due to hardware failures, software glitches, or unforeseen disasters) is met with frustration and loss. It is in this landscape that the concepts of high availability (HA) and disaster recovery (DR) emerges as a linchpin of resilience and continuity.

With the growing demands of modern businesses, it's essential for critical applications to remain constantly available and for systems to be fault-tolerant. However, implementing such fault-tolerant systems often incurs significant costs. Therefore, there's a need for a solution that delivers these capabilities while remaining cost-effective.

A high availability solution ensures that the failure of any individual component does not impact the accessibility of the application or its data for users. This is accomplished by eliminating single points of failure, thereby reducing or masking both planned and unplanned downtime.

When HA measures fall short, DR steps in. Typically, DR is reserved for large-scale failures, such as a complete site failure or data corruption due to cyber attacks. DR often involves manual processes, due to the high stakes involved. It involves straightforward actions like restoring from a backup or executing a site failover, but it can also become highly intricate, including analyzing the database log to maintain logical consistency and deciding individually whether to apply particular database operations or not.

Figure 2-12 illustrates failover capability, where if one server fails, another seamlessly takes over its operations.

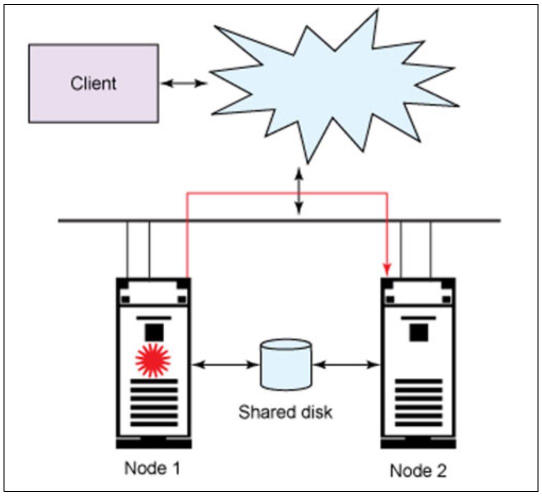


Figure 2-12 Failover capacity for high availability

The configuration depicted in Figure 2-12 on page 67 offers a high availability solution within a single site. However, in the event of a site-level outage caused by an electrical power failure, fire, flood, or other natural disaster, additional plans are necessary to recover your data and maintain application operations. Figure 2-13 demonstrates an HA/DR solution using a three-site architecture, consisting of two on-premise sites managed by the client and an additional failover site with servers and storage systems hosted in the IBM Cloud. Assuming Site 1 and Site 2 are located within the same data center or city, applications can be restarted at Site 2 with minimal or no downtime if Site 1 experiences a planned or unplanned outage. If both Site 1 and Site 2 are affected, applications can be restarted at the remotely located Site 3.

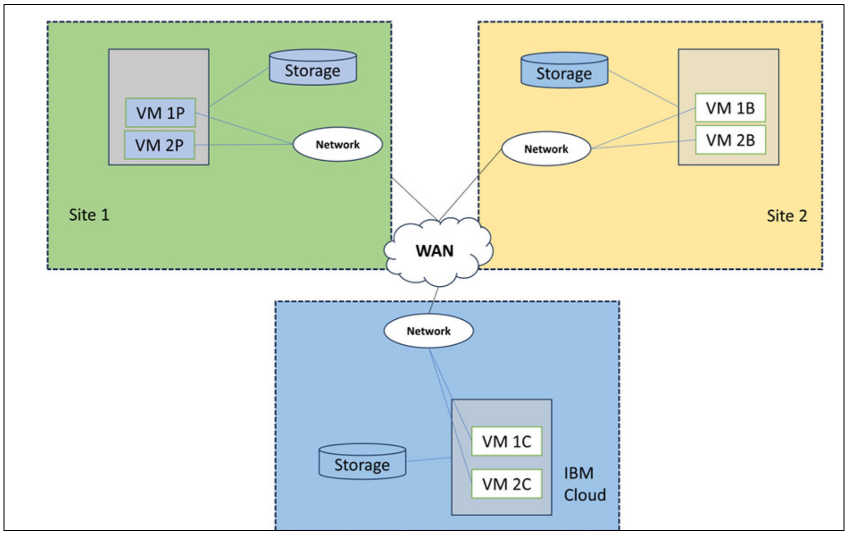


Figure 2-13 Full HA/DR solution utilizing three sites with storage replication

2.5.1 IBM Solutions for HA/DR

IBM Power (including IBM Power or the Cloud utilizing IBM Power Virtual Server) provide an entire product line of HA and DR solutions. Figure 2-14 provides an overview of the IBM HA/DR capabilities on IBM Power systems.

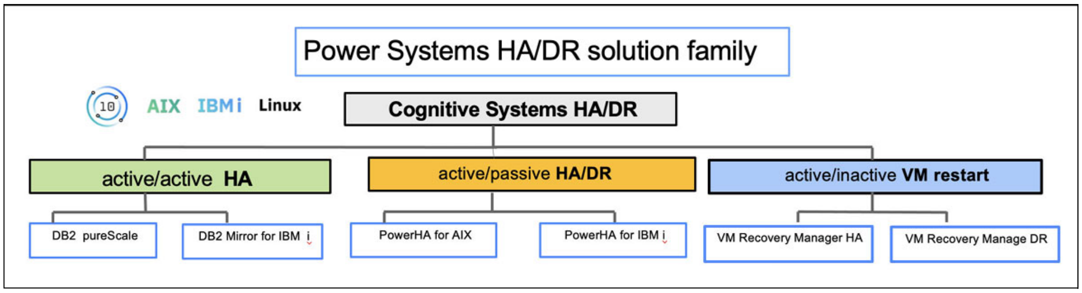


Figure 2-14 IBM Power Systems virtual server HA/DR solution family.

- PowerHA SystemMirror

  The IBM PowerHA SystemMirror family of solutions is optimized for mission-critical applications where the total annual downtime for both planned and unplanned outages is zero or near-zero. The PowerHA System mirror product line covers all outage types, both software and hardware. There is at least one active OS on each of the nodes in the cluster, which enables software updates on a system other than the production node. PowerHA SystemMirror covers both data center and multi-site configurations. To drive total outage time for both planned and unplanned events to near zero, this solution is the one to deploy. For more information, see2.5.2, “IBM PowerHA SystemMirror” on page 70.

- Virtual Machine Recovery Manager

  Virtual Machine Recovery Manager (VMRM) solutions can be best understood by first understanding Live Partition Mobility (LPM). A set of logical partitions (LPARs) is virtualized using IBM PowerVM and Virtual I/O Server (VIOS) to enable a partition to be moved for a firmware or hardware maintenance event by using LPM. If that VM fails, it can be restarted on another server in the cluster. For DR operations, those same VMs are replicated using storage area network (SAN) storage at the secondary location. For more information, see IBM VM Recovery Manager for Power.

- Active-Active Solutions

  The IBM active-active solutions are IBM Db2 pureScale® (which supports AIX), and IBM Db2 Mirror for i. In both cases, the solutions are classified as active-active, but they are achieved using different approaches.

  – Db2 pureScale provides an active/active solution using a shared Db2 cluster configuration with distributed lock management to enable multiple application servers to simultaneously access the shared database.

  – IBM Db2 Mirror for i offers continuous availability for mission-critical applications by synchronously mirroring database updates between two separate nodes using Remote Direct Memory Access (RDMA) over Converged Ethernet (RoCE) network.

  Active-active solutions are available for other database solutions as well. For example using Oracle RAC and Oracle Data Guard to provide highly available Oracle Database solutions.

Table 2-1 highlights the differences between IBM Power HA/DR 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. Therefore, there is no application restart upon a node outage. Certain types enable read-only access from secondary nodes.	OS clustering: one OS in the cluster has access to the production data. Multiple active OS instances on all nodes in the cluster. Application is restarted on a secondary node upon outage of a production node.	VM Clustering: one VM in a cluster pair has access to the data, one logical OS, and two physical copies. OS and applications must be restarted on a secondary node upon a primary node outage event. LPM enables the VM to be moved non-disruptively for a planned outage event.
Outage Types	► SW, HW, HA, planned, unplanned ► RTO 0, limited distance	► SW,HW,HA,DR,planned, unplanned ► RTO>0, multi-site	► HW,HA,DR, planned, unplanned ► RTO>0, multi-site
OS integration	Inside the OS	Inside the OS	OS agnostic
RPO	Sync mode only	Sync/Async	Sync/Async
RTO	Zero	Fast (minutes)	Fast Enough (VM Reboot)
Licensing	N+N[a]	N+1 licensing	N+0 licensing
IBM Solution	Db2 pureScale, Db2 Mirror[b]	PowerHA, Red Hat HA, Linux HA	VMRM HA/DR

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 high availability solution that integrates both storage and availability needs within a single, streamlined configuration and user-friendly interface. IBM Power remains dedicated to developing and delivering solutions that enhance the resilience of your IT environment.

IBM PowerHA technology is offered in two editions – Standard and Enterprise– for both IBM AIX and IBM i. Starting with IBM i 7.5, PowerHA for IBM i will be available 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 offers a cost-effective commercial computing solution that enables rapid recovery of mission-critical applications in the event of hardware or software failures.

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

This high availability solution leverages a combination of custom software and industry-standard hardware to minimize downtime by swiftly restoring services following a failure in the system, a component, or an application. While not instantaneous, service recovery typically occurs within 30 to 300 seconds.

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

PowerHA SystemMirror helps you with the following:

- The PowerHA SystemMirror planning process and documentation include tips and advice on the best practices for installing and maintaining a highly available PowerHA SystemMirror cluster.
- Once the cluster is operational, PowerHA SystemMirror provides the automated monitoring and recovery for all the resources on which the application depends.
- PowerHA SystemMirror provides a full set of tools for maintaining the cluster while keeping the application available to clients.

PowerHA SystemMirror allows you to:

- Quickly and easily setup a basic two-node cluster by using the typical initial cluster configuration SMIT path or the application configuration assistants (Smart Assists).
- Test your PowerHA SystemMirror configuration by using the Cluster Test Tool. You can evaluate how a cluster behaves under a set of specified circumstances, such as when a node becomes inaccessible, a network becomes inaccessible, and so forth.
- Ensure high availability of applications by eliminating single points of failure in a PowerHA SystemMirror environment.
- Leverage high availability features available in AIX.
- Manage how a cluster handles component failures.
- Secure cluster communications.
- Monitor PowerHA SystemMirror components and diagnose problems that might occur.

More information on IBM PowerHA SystemMirror for AIX is available at https://www.ibm.com/docs/en/powerha-aix/7.2.x.

IBM PowerHA SystemMirror for IBM i

IBM PowerHA SystemMirror for i delivers comprehensive, integrated clustering solutions for both high availability (HA) and disaster recovery (DR). As a built-in extension of the IBM i operating system, PowerHA offers robust resiliency for the environment, applications, and data—ensuring 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 HA and DR management across a wide range of IBM i-based systems and storage configurations. These include:

- Geographic Mirroring

A Geographic mirror is a system-based solution capable of providing either synchronous or asynchronous transmission options. This replication solution is not dependent on any particular type of storage technology.

- Metro Mirroring

  A Metro mirror solution is a synchronous copy services function that manages supported external storage units. This provides hardware-level replication of Independent auxiliary storage pools (IASPs) from one cluster node to another.

- Global Mirroring

  The Global mirror is an asynchronous copy services function that manages supported external storage units. This provides hardware-level replication of Independent auxiliary storage pools (IASPs) from one cluster node to another.

- Switched logical units

  Logical unit (LUN) switching switches a supported external storage unit connection between two systems in a cluster to move an IASP from one cluster node to another.

- IBM DS8000 HyperSwap® and PowerHA

  HyperSwap is a full system solution that allows for logical units that are mirrored between two IBM System Storage DS8000 units. PowerHA provides an interface to manage either Full System HyperSwap or HyperSwap configured at the IASP level.

- IBM FlashCopy® and PowerHA

  FlashCopy is a function of the IBM System Storage server. FlashCopy provides a fast point-in-time copy of data that can be stored, used, or brought online on a separate partition or system. FlashCopy can be used together with other data resiliency technologies and managed with PowerHA.

These solutions can be used separately or combined together based on the requirements of your organization.

For additional information about IBM PowerHA System Mirror for i, visit the PowerHA Wiki.

2.5.3 Virtual Machine Recovery Manager

Virtual Machine Recovery Manager (VMRM) is an automated solution that implements recovery of your partitions by using restart technology. VMRM relies on an out-of-band monitoring and management component that restarts the VMs on another server when the host infrastructure fails. There are multiple deployment options that are provided by VMRM.

Depending on your requirements, VMRM can provide an HA solution within a single site or two sites within metro distances by using VMRM HA. To support your DR requirements across sites that are farther apart, VMRM DR provides a solution for managing workloads between your production or primary site and your secondary (backup) or DR site. For customers that want more flexibility on where the workloads are recovered, two more options are supported: HADR and HADRHA.

VMRM should be differentiated from a clustering technology that deploys redundant hardware and software components for a near real-time failover operation when a component fails. The VMRM HA is ideal to ensure HA for many VMs, and if it meets your RPO requirements, it is simpler to manage than cluster environments because it does not have the complexities of clustering.

Because VMRM HA is based on SRR technology, it is OS-independent and can be used with IBM AIX, IBM i, or Linux. This simplified management capability is extended to a second site for DR solutions by using VMRM DR. The additional options of VMRM HADR and VMRM HADRHA are extensions and combinations of the VMRM HA and VMRM DR solutions to provide integrated HADR capabilities.

VMRM HA

HA management is a critical feature of business continuity plans. Any downtime to the software stack can result in loss of revenues and disruption of services. VMRM HA for Power is a HA solution that is easy to deploy, and provides an automated solution to recover the VMs, also known as LPARs. The VMRM HA solution implements recovery of the VMs based on the VM restart technology (SRR). The VM restart technology relies on an out-of-band monitoring and management component that restarts the VMs on another server when the host infrastructure fails. The VM restart technology is different from the conventional cluster-based technology that deploys redundant hardware and software components for a near real-time failover operation when a component fails. The VMRM HA solution is ideal to ensure HA for many VMs. Also, the VMRM HA solution is easier to manage because it does not have clustering complexities.

VMRM DR

DR of applications and services is a key component to provide continuous business services. The VMRM DR for Power solution is a DR solution that is simple to deploy and provides automated operations to recover the production site. The VMRM DR solution is based on the IBM Geographically Dispersed Parallel Sysplex® (IBM GDPS®) offering concept that optimizes the usage of resources. This solution does not require the deployment of backup VMs for DR, so the VMRM DR solution reduces the software license and administrative costs. The VMRM DR solution is based on the VM restart technology across two sites. The VM restart-based HADR solution relies on an out-of-band monitoring and management component (provided by VMRM DR) that restarts the VMs on other hardware when the host infrastructure fails.

VMRM HADR

VMRM HADR provides more HA features in addition to the VMRM DR solution. This solution provides more flexibility about where your applications can run after a failure recovery by supporting the recovery of your applications at your primary and secondary sites. The important HA features of the HADR type of deployment of the VMRM DR solution include the following:

- LPM support within a site Support for VM failure and host failure within a site
- Application failover support within a site
- Fibre Channel (FC) adapter failure within a site
- Network adapter failure within a site
- Monitoring CPU and memory usage within a site
- Move operation and failover rehearsal operation across sites and within a site
- DR support from one site to another site.

VMRM HADRHA

The VMRM HADRHA solution adds extra HA features to the VMRM DR solution and the VMRM HADRR solution. The HADRHA solution allows recovery of your applications within the same site at both the primary site and the secondary site.

The important HA features of the HADR type of deployment of the VMRM DR solution includes the following:

- LPM support within a site
- Support for VM failure and host failure within a site
- Application failover support within a site

- FC adapter failure within a site
- Network adapter failure within a site
- Monitoring CPU and memory usage within a site
- Move operation and failover rehearsal operation across sites and within a site
- LDR support from one site to another site

2.5.4 Linux High Availability Clustering

IBM VM Recovery Manager (VMRM) utilizes VM restart technology, making it OS agnostic and capable of providing high availability (HA) and disaster recovery (DR) for Linux virtual machines. Previously, IBM offered a version of IBM PowerHA Recovery Manager that supported Linux, but this product has been discontinued and is no longer supported. The discontinuation was due to the development of Pacemaker by the open-source community, which offers a comprehensive Linux HA clustering solution. Pacemaker, along with CoroSync – responsible for managing communication between cluster nodes and cluster membership – is widely used across multiple distributions to deliver a full-featured cluster management solution for Linux systems.

Pacemaker

Pacemaker is an open-source high-availability cluster resource manager designed to manage and maintain the integrity of services running on a set of hosts, known as a cluster. It detects and recovers from host- and application-level failures, ensuring minimal downtime and preserving data integrity. Pacemaker supports various redundancy configurations, including active/passive and N+1 setups, and can manage practically any application that can be scripted. Its ability to handle complex dependencies between services, such as ordering and collocation, makes it a versatile solution for maintaining high availability in Linux environments.

Pacemaker's architecture is built around the concept of resources, which are the services that need to be kept highly available. Resource agents, which are scripts or operating system components, start, stop, and monitor these resources based on a set of parameters. Pacemaker also includes fencing capabilities, known as STONITH (Shoot The Other Node In The Head), which ensure that faulty nodes are isolated to prevent data corruption. This is achieved through devices like intelligent power switches or network switches that cut power or network access to the target node.

Pacemaker is widely supported across various Linux distributions, including Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES) 3. Both distributions integrate Pacemaker as part of their high-availability solutions, providing users with robust tools to configure and manage clusters. Red Hat, for instance, offers detailed documentation and support for setting up Pacemaker clusters to ensure high availability of critical applications[1]. SUSE also provides comprehensive guides and support for deploying Pacemaker in enterprise environments, ensuring that services remain available even during failures.

The collaborative development of Pacemaker by the ClusterLabs community, including contributions from Red Hat and SUSE, has led to its widespread adoption and deployment in critical environments. This widespread use underscores Pacemaker's reliability and effectiveness as a high-availability cluster manager for Linux systems.

[1] https://www.redhat.com/en/blog/rhel-pacemaker-cluster

2.5.5 Additional information on HA and DR solutions

These IBM Redbooks provide further details on HA and DR solutions in IBM Power environments:

- *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, fueled by the rapid advancements in artificial intelligence (AI). AI-powered coding assistants are at the forefront of this shift, enhancing productivity, improving code quality and streamlining workflows. These intelligent tools cannot only augment human capabilities but also redefine best practices, accelerate development cycles and drive greater efficiency.

According to Gartner Magic Quadrant for AI Code Assistants[2], 90% of enterprise software engineers will use AI code assistants by 2028, a sharp increase from less than 14% in early 2024. Also, Markets and Markets research[3] states that the global market for AI Code Tools Market is projected to 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% during the forecast period.

AI will influence the future of software development and every stage of the development lifecycle. From code generation and unit test automation to code transformation, bug detection and autonomous fixes, AI-driven assistants and intelligent agents can revolutionize the way developers work. These technologies will not only enhance efficiency but also establish new industry standards, with the potential to make software development faster, more reliable and increasingly automated.

The fast pace of AI development has accelerated coding assistants to evolve from emerging technologies to tools within the developer's toolkit. AI-powered assistants, such as IBM watsonx Code Assistant™, provide developers with intelligent code suggestions, automated bug troubleshooting and code optimization, empowering them 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, helping boost productivity and empowering development teams of all skill levels to support business innovation. It is tailored to accelerate workflows across Python, Java, C®, C++, Go, JavaScript, Typescript and more, with gen AI assistance that integrates directly into their IDE. IBM watsonx Code Assistant™ helps improve business agility and reduce technical debt with a customized experience that automates the end-to-end modernization of Java applications and runtime environments.

Powered by IBM Granite® code models, watsonx Code Assistant can help increase developer productivity with context-aware assistance in the IDE, as well as with its chat function, helping generate, complete, transform, explain, document, review and test code. The ability to deploy the

[2] https://www.gartner.com/doc/reprints?id=1-2IKO4MP6&ct=240819&st=sb&trk=ef359388-b2c8-4403-bbc0-768ca2834bfb&sc_channel=el

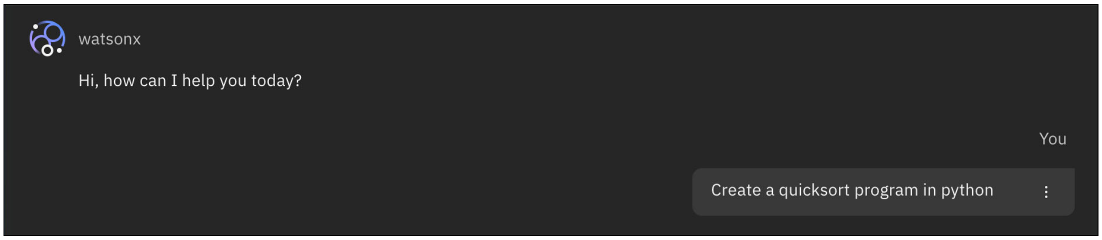
[3] https://www.marketsandmarkets.com/Market-Reports/ai-code-tools-market-239940941.html

solution on the cloud or on premise with IP indemnification promotes trust and transparency in your code.

WCA Features

- Accelerate code generation:

  WCA can generate code suggestions, unit tests, and even entire functions or methods, allowing developers to work more efficiently and focus on higher-level tasks. Write high-quality code aligned with established conventions using simple, natural language prompts - regardless of your experience level. Figure shows the initial prompt for watsonx where the user requests a python program for a quicksort.



watsonx

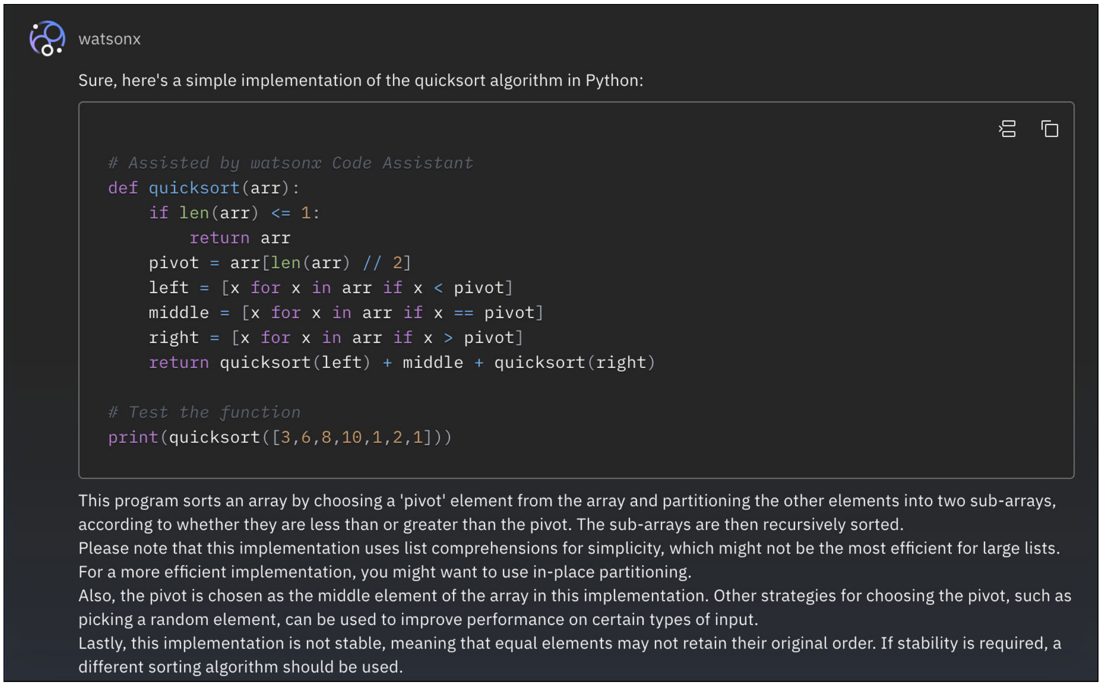
Hi, how can I help you today?

You

Create a quicksort program in python

Figure 2-15 watsonx request for python code

Figure 2-16 is the response with the code and a code description.



watsonx

Sure, here's a simple implementation of the quicksort algorithm in Python:

```
# 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.

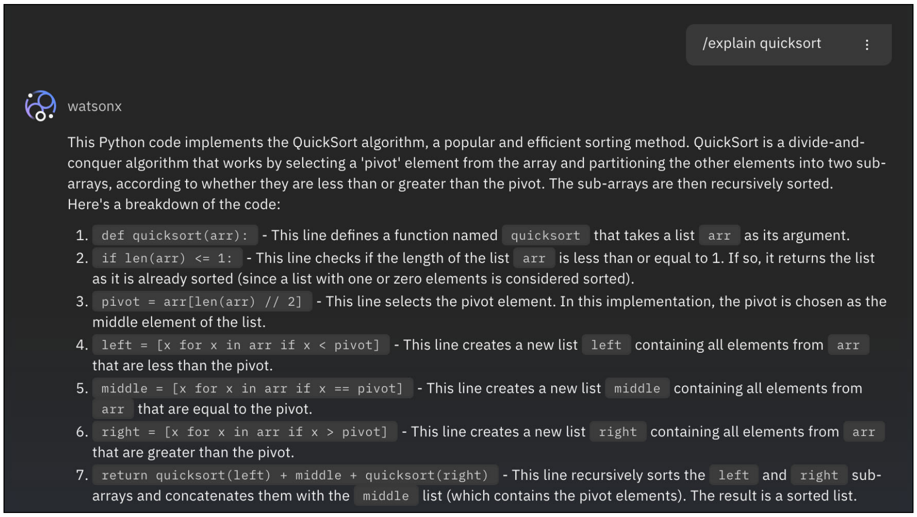
Figure 2-16 Response to request for python code

- Code Explanation

  WCA uses generative AI to analyze and summarize code, providing insights into its functionality and purpose. This feature enhances transparency and facilitates better collaboration among development team.

Figure 2-18 shows the code explanation generated by watsonx.

.



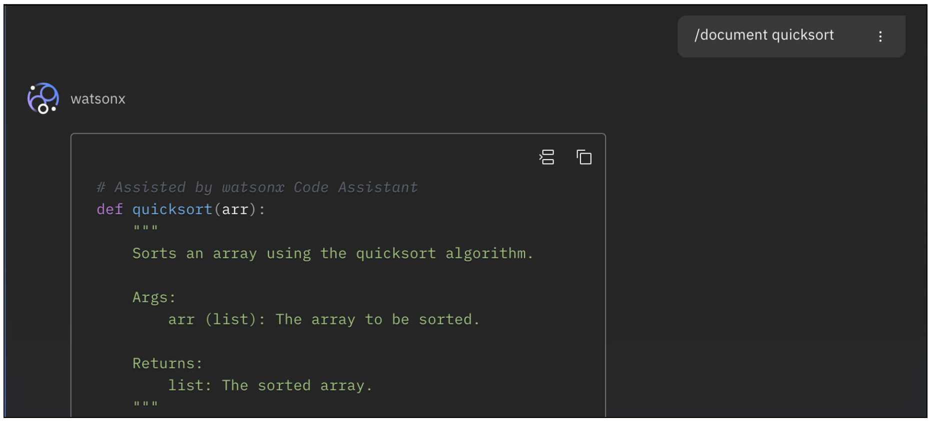
This Python code implements the QuickSort algorithm, a popular and efficient sorting method. QuickSort is a divide-and-conquer algorithm that works by selecting a 'pivot' element from the array and partitioning the other elements into two sub-arrays, according to whether they are less than or greater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

- Documenting code

  WCA can generate comment lines that document what the code does, making it easier for developers to understand and maintain their codebase. Figure 2-17 shows a the comments added for documenting the code.



```
# 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:

  WCA's ability to generate unit tests and explain code 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 cycle, driving IT efficiency and scalability with trusted AI. It simplifies content creation with natural language Playbook generation, detailed explanations, and a customizable model that provides tailored recommendations, helping to ensure a confident and enhanced experience for developers at any skill level.

The code assistant provides:

- Playbook generation and code explanation

  Reduce the time needed to generate complete Ansible Playbooks from natural language and receive detailed explanations for each task.

- Model tuning

  Tailor the model further with your existing Ansible Playbook content for more personalized code recommendations that are similar and fit for your enterprise.

- Ansible content generation

  Simplify the process of generating Ansible Playbooks by using natural language inputs in the Ansible Task description.

- Content source matching

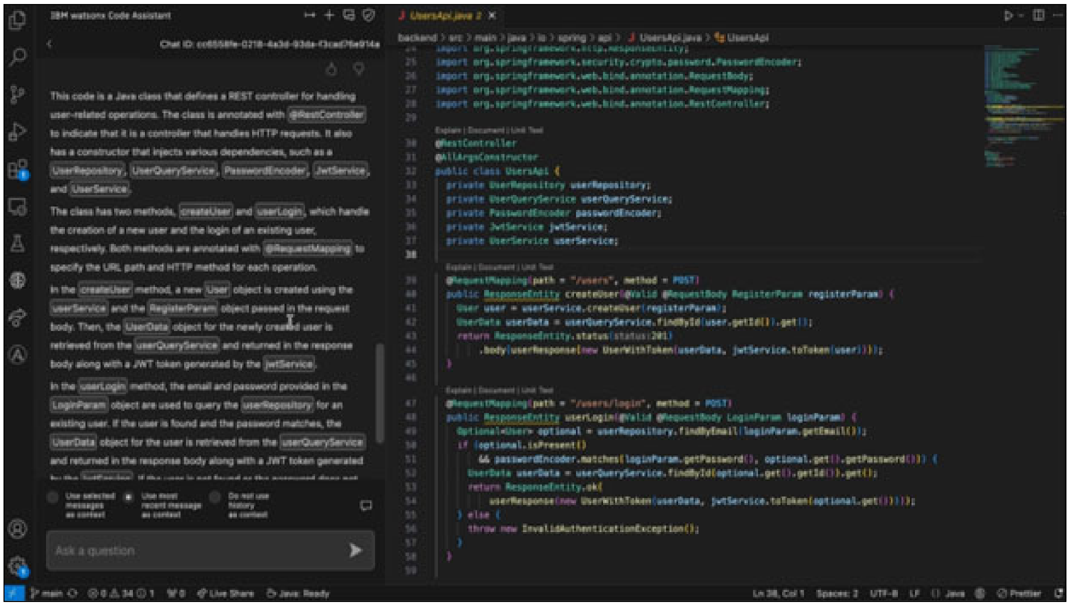
  Make informed decisions about which suggestions to accept or reject based on their understanding of why those suggestions were made.

2.6.3 Benefits of AI Code Assistants

The impact of AI on software development can extend far beyond automating routine tasks. Beyond handling repetitive tasks with machine learning capabilities, AI coding assistants can learn from vast codebases, offering contextually relevant suggestions that align with the project's architecture, coding style and best practices. This can result in not only syntactically correct code but also a more holistic approach to software development, making it easy for developers to streamline workflows, enhance code maintainability and accelerate software delivery.

- Boost developer productivity

  One of the most immediate benefits of AI coding assistants is the increase in developer productivity. Based on an internal IBM test, IBM developers using IBM watsonx Code Assistant projected that they could see a 90% time savings on code explanation, 59% time reduction on documentation and 38% time reduction in code generation and testing. Figure 2-19 shows an example of Java code generation using WCA


IBM watsonx Code Assistant
Chat ID: cc6558fe-0218-4a3d-93da-f3cad76e914a

This code is a Java class that defines a REST controller for handling user-related operations. The class is annotated with @RestController to indicate that it is a controller that handles HTTP requests. It also has a constructor that injects various dependencies, such as a UserRepository, UserQueryService, PasswordEncoder, JwtService, and UserService.

The class has two methods, createUser and userLogin, which handle the creation of a new user and the login of an existing user, respectively. Both methods are annotated with @RequestMapping to specify the URL path and HTTP method for each operation.

In the createUser method, a new User object is created using the userService and the RegisterParam object passed in the request body. Then, the UserData object for the newly created user is retrieved from the userQueryService and returned in the response body along with a JWT token generated by the jwtService.

In the userLogin method, the email and password provided in the LoginParam object are used to query the userRepository for an existing user. If the user is found and the password matches, the UserData object for the user is retrieved from the userQueryService and returned in the response body along with a JWT token generated by the jwtService. If the user is not found or the password does not

○ Use selected messages as context
● Use most recent message as context
○ Do not use history as context
Ask a question
UsersApi.java 2
backend > src > main > java > io > spring > api > UsersApi.java > UsersApi

```java
import org.springframework.http.ResponseEntity;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

Explain | Document | Unit Test
@RestController
@AllArgsConstructor
public class UsersApi {
  private UserRepository userRepository;
  private UserQueryService userQueryService;
  private PasswordEncoder passwordEncoder;
  private JwtService jwtService;
  private UserService userService;

  Explain | Document | Unit Test
  @RequestMapping(path = "/users", method = POST)
  public ResponseEntity createUser(@Valid @RequestBody RegisterParam registerParam) {
    User user = userService.createUser(registerParam);
    UserData userData = userQueryService.findById(user.getId()).get();
    return ResponseEntity.status(status:201)
        .body(userResponse(new UserWithToken(userData, jwtService.toToken(user))));
  }

  Explain | Document | Unit Test
  @RequestMapping(path = "/users/login", method = POST)
  public ResponseEntity userLogin(@Valid @RequestBody LoginParam loginParam) {
    Optional<User> optional = userRepository.findByEmail(loginParam.getEmail());
    if (optional.isPresent()
        && passwordEncoder.matches(loginParam.getPassword(), optional.get().getPassword())) {
      UserData userData = userQueryService.findById(optional.get().getId()).get();
      return ResponseEntity.ok(
          userResponse(new UserWithToken(userData, jwtService.toToken(optional.get()))));
    } else {
      throw new InvalidAuthenticationException();
    }
  }
```

Figure 2-19 Java code generated by WCA

- Elevate code quality

  IBM watsonx Code Assistant is a critical tool in enhancing code quality. With real-time feedback and suggestions, it helps developers proactively identify potential issues and detect bugs early in the development process. Its ability to offer best practice and optimize code structure can lead to fewer errors, enhanced maintainability and strengthened software reliability and security.

  Poor code quality can result in bugs, security vulnerabilities and costly maintenance. AI-powered coding assistants can help mitigate these risks by identifying issues early, so software remains robust, scalable and easier to maintain. By integrating AI-driven insights into the development workflow, organizations have an opportunity to improve code quality, reduce technical debt and accelerate software delivery.

- Foster collaboration and knowledge sharing

  AI coding assistants also play a pivotal role in fostering collaboration, particularly in distributed and remote development teams. With features—such as real-time code suggestions, shared knowledge bases, AI-driven feedback loops and output sharing—tools such as IBM watsonx Code Assistant are designed to enhance communication among team members. These capabilities allow teams, especially those working across different time zones, to collaborate effectively and align on best practices.

  By automating mundane tasks and providing support for more complex tasks, coding assistants can also help standardize code quality and practices so that all team members are on the same page, regardless of their experience level or geographic location. This consistency can help promote collaboration and accelerate the development process.

- Lower barrier for new software developer

  Perhaps one of the most promising outcomes of AI-powered coding assistants is their potential to lower the barrier to entry for new developers. By automating routine tasks, such as code snippet generation and code completion, and providing code reviews and intelligent suggestions, tools such as IBM watsonx Code Assistant can make it easier for beginners to learn new programming languages and contribute to projects with greater confidence.

  For those just starting their coding journey, AI coding assistants with natural language processing capabilities provide real-time feedback, offer helpful resources and suggest improvements, helping to make the learning curve less daunting and more engaging.

Looking ahead, the potential for AI-driven development is immense. As AI technology continues to evolve, coding assistants will continue to become even more advanced, moving beyond code suggestions to anticipate a developer’s needs, understand project context and provide proactive assistance.

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

3

Client examples and use cases

A modernization journey on IBM Power Systems typically involves a structured approach to enhance infrastructure, applications, and processes to align with modern technologies, business goals, and evolving IT needs. As technology evolves, organizations look for ways to integrate newer technologies, optimize performance, and future-proof their infrastructure.

One of the challenges in starting on a modernization journey is that clients often do not know what their options are and how to get started. In this chapter we provide some examples of how IBM clients (including IBM's internal services) modernized their Power infrastructure and applications.

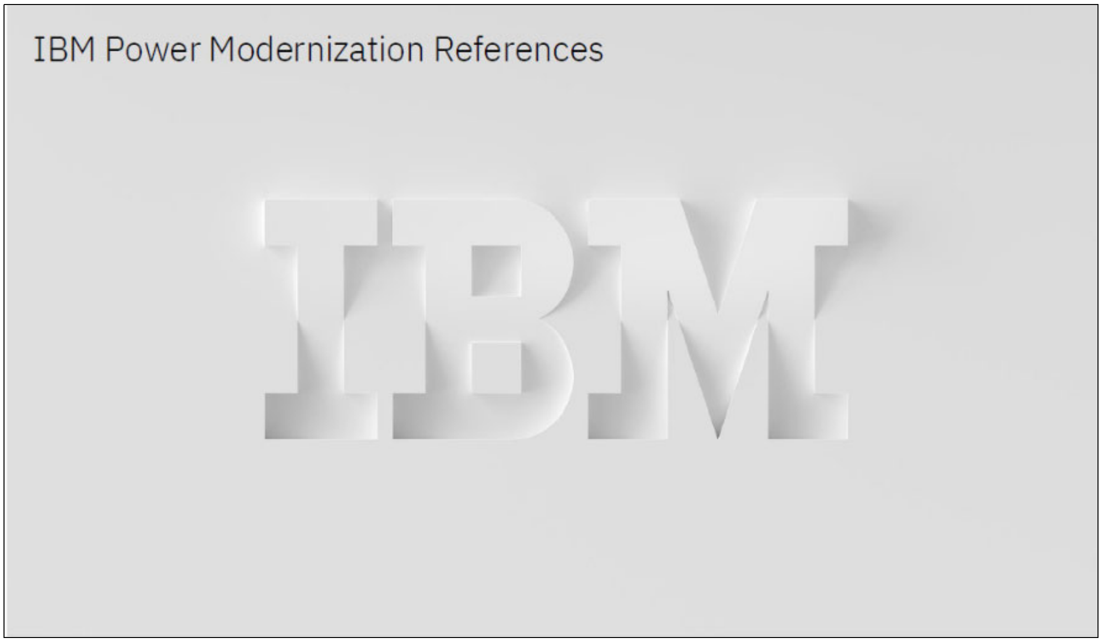
The following topics are covered in this chapter:

© Copyright IBM Corp. 2025.

3.1 IBM Power Modernization References

Companies worldwide, across multiple industries, are finding an edge over competitors with technology running on IBM Power. Modernizing mission-critical applications on a tried and trusted cloud-native platform with Red Hat OpenShift and flexible capacity on-demand to deliver industry leading applications faster. Bettering performance from IT infrastructure drives down energy costs to achieve a better TCO while keeping up with green IT regulations.

Implementing industry leading isolation and integrity for ultra secure cloud native containerized applications to reduce risks while propelling hybrid-cloud transformation. Depending on a trusted pre-sales and post-sales support system to fast-track successful co-creation, co-execution and co-management. Find examples of how choosing IBM Power translates into tangible outcomes at this IBM Power Modernization References document.



We created the above link and shared the content specifically for the audience of this Redbook publication. We will continue to add additional references to dynamically share additional use case and references, so please check back regularly for further updates.

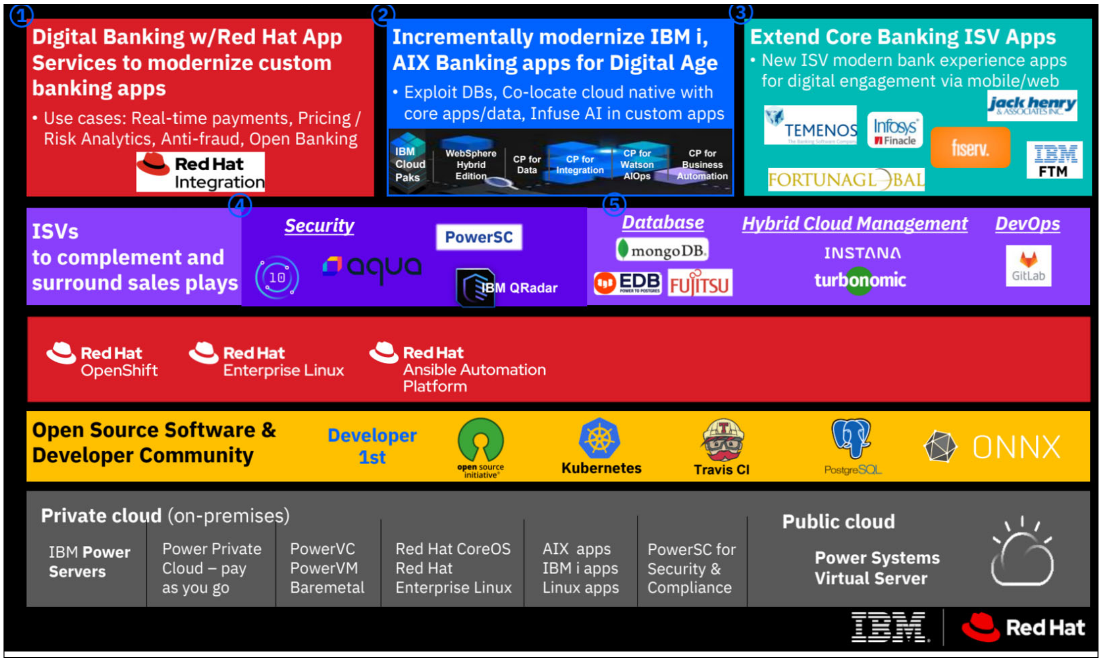
3.2 Banking solutions

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 for modernizing their IT environments. Hybrid cloud enables banks to operate with greater agility, allowing them to swiftly respond to and scale with customer demand. It also facilitates the rapid adoption of technologies that enhance efficiency, security, and sustainability.

IBM and Red Hat offer a robust hybrid cloud solution for banking modernization, utilizing Red Hat OpenShift on IBM Power Systems to create scalable, secure, and agile digital banking applications. This solution enables three distinct use cases:

1. Utilizing New Cloud Services:
   - Modernize to Real-Time Payments: Meet new open banking regulations, add faster payment methods, and simplify fund transfers across financial institutions.
   - Enhance Anti-Fraud Activities: Gain rapid access to data and monitor with adaptive rules to detect suspicious activity.
   - Drive Business Innovation: Use new tools and capabilities to create and market offers, build new customer experiences, and grow your business.
2. Incrementally Modernizing Legacy Applications:
   - Modernize applications running on AIX and IBM i platforms.
3. Extending Core Banking ISV Applications:
   - Support new digital engagement patterns.

Figure 3-1 illustrates the IBM and Red Hat ecosystem that enables your modernization journey.



1 Digital Banking w/Red Hat App Services to modernize custom banking apps

- Use cases: Real-time payments, Pricing / Risk Analytics, Anti-fraud, Open Banking

2 Incrementally modernize IBM i, AIX Banking apps for Digital Age

- Exploit DBs, Co-locate cloud native with core apps/data, Infuse AI in custom apps

3 Extend Core Banking ISV Apps

- New ISV modern bank experience apps for digital engagement via mobile/web

4 ISVs to complement and surround sales plays

Security

5 Database

Hybrid Cloud Management

DevOps

Open Source Software & Developer Community

Developer 1st

Private cloud (on-premises)

IBM Power Servers

Power Private Cloud – pay as you go

PowerVC PowerVM Baremetal

Red Hat CoreOS Red Hat Enterprise Linux

AIX apps IBM i apps Linux apps

PowerSC for Security & Compliance

Public cloud

Power Systems Virtual Server

Figure 3-1 Banking use cases

For one approach to banking modernization refer to Banking and Finance Modernization with MongoDB and Red Hat OpenShift on IBM Power.

3.3 IBM CIO – Hybrid by Design

IBM has built a world class hybrid cloud platform designed using Red Hat OpenShift in our cloud, on IBM Power and on IBM Z Mainframes. We have achieved over $3.5 B in productivity gains in the last two years, with $2 B of those coming from AI and automation. We have transformed our operating model to eliminate complexity, simplify end-to-end workflows, automated manual tasks and are embedding AI on our hybrid cloud.

Central to this is the principle of “drinking our own champagne” – that is to say, testing out our technologies on ourselves. We call this our ***“Client Zero”*** approach**.**

3.3.1 IBM CIO Cirrus

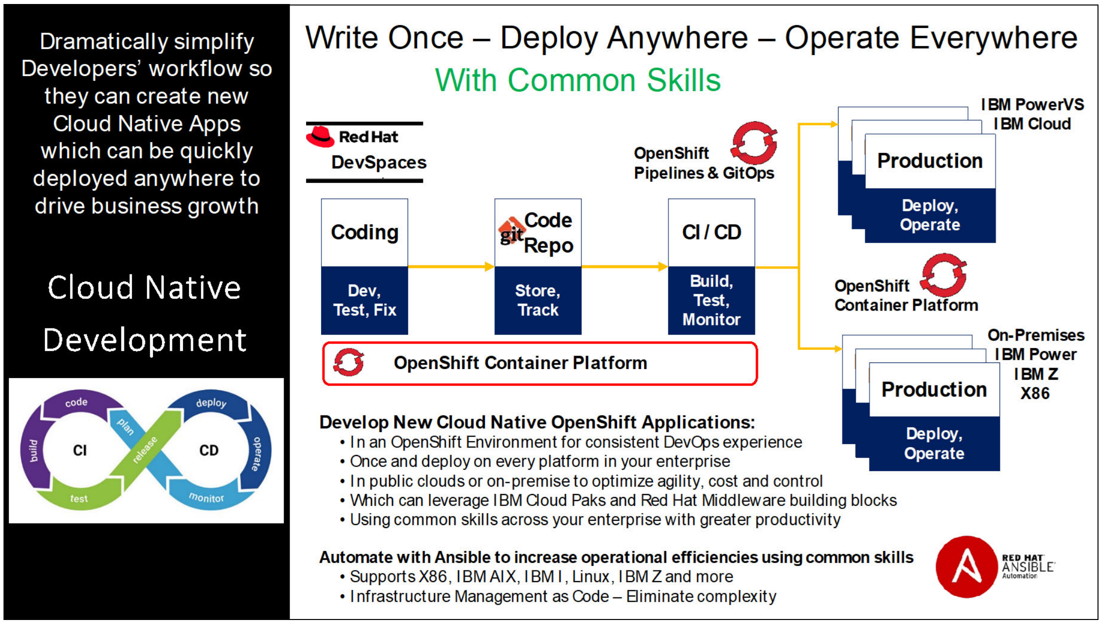
Cirrus, our internal hybrid cloud platform, is built with Red Hat and IBM technologies. Modern and easy to use, Cirrus provides the speed, scale, security, and simplicity needed for IBM's digital business experiences – connecting data, information, and people to bring IBM into the future.

The IBM CIO Hybrid Cloud is the engine of digital transformation, bringing security-rich speed, scale, and simplicity to the IBM Chief Information Officer (CIO) Organization’s operation features that are not available when using exclusively public cloud or exclusively private cloud. It provides the velocity for the company’s digital application teams to focus on what is most important: IBM’s digital business.

The three principles of Cirrus are:

- Hybrid – One consistent experience across all platforms.
- Integrated – Common CI/CD pipeline and operating environment.
- Open – Leverage open-source and standard solutions.

Core to our “Hybrid by Design” approach was to leverage Red Hat OpenShift to create a common CI/CD (Continuous Integration/Continuous Deployment or Delivery) pipeline that worked on x86, IBM Power (AIX, IBM I, and Linux), and IBM Z environments. This approach is shown in Figure 3-2.



Dramatically simplify Developers’ workflow so they can create new Cloud Native Apps which can be quickly deployed anywhere to drive business growth

Cloud Native Development

Write Once – Deploy Anywhere – Operate Everywhere

With Common Skills

Develop New Cloud Native OpenShift Applications:
- In an OpenShift Environment for consistent DevOps experience
- Once and deploy on every platform in your enterprise
- In public clouds or on-premise to optimize agility, cost and control
- Which can leverage IBM Cloud Paks and Red Hat Middleware building blocks
- Using common skills across your enterprise with greater productivity

Automate with Ansible to increase operational efficiencies using common skills
- Supports X86, IBM AIX, IBM I, Linux, IBM Z and more
- Infrastructure Management as Code – Eliminate complexity

Figure 3-2 IBM CIO Cirrus goals

IBM is a company with more than 250,000 employees and annual revenues that exceed USD 50 billion. It takes thousands of internal applications to manage all aspects of the company’s diverse portfolio. IBM has applications for chip design, sales, marketing, management, accounting, customer support and many other functions. Some of these applications are as old as the computer industry. Others support the latest developments in quantum computing.

IBM’s application teams were facing difficult problems:

- With over 5,000 applications and 74 datacenters.
- Used by 250,000+ users in over 170 countries.

- Hosting and operational costs that were not controlled.
- Keeping current with the latest security patches.
- Meeting availability and disaster recover objectives.
- Satisfying corporate security requirements.
- Making minor changes was taking months.

Our ***"Client Zero"*** success has enabled IBM to leverage a consolidated hybrid cloud supporting our AI and Automation solutions to transform our enterprise productivity.

Hybrid by Design

The CIO Organization envisioned a "hybrid by design" cloud platform – a single platform on which digital business application components would 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," says Matt Lyteson, CIO, VP Technology Platforms Transformation at IBM.

Figure 3-3 illustrates the IBM CIO vision

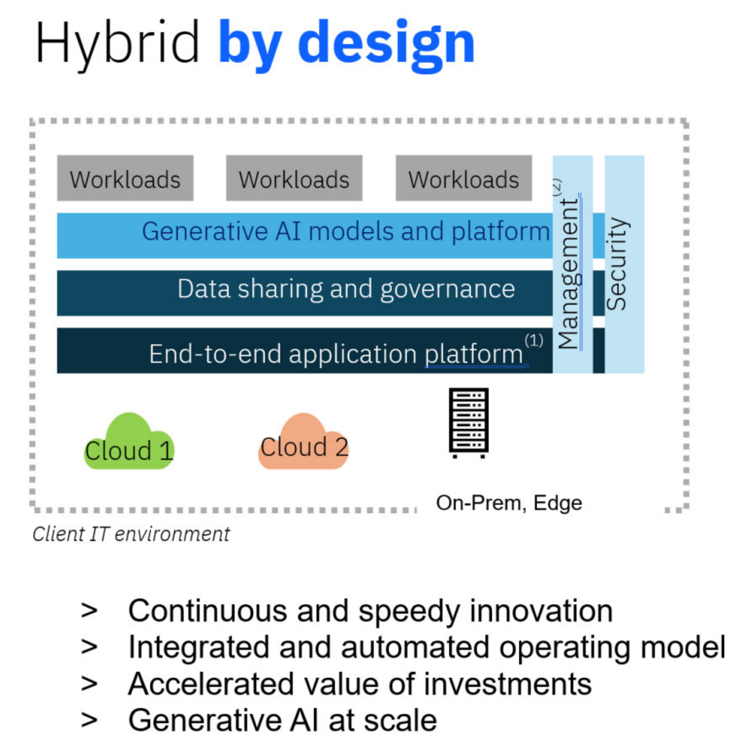

Hybrid by design
Workloads
Workloads
Workloads
Generative AI models and platform
Data sharing and governance
End-to-end application platform(1)
Management(2)
Security
Cloud 1
Cloud 2
On-Prem, Edge
Client IT environment
> Continuous and speedy innovation
> Integrated and automated operating model
> Accelerated value of investments
> Generative AI at scale

Figure 3-3 Hybrid by design concepts

Platform engineering for IBM CIO hybrid cloud at scale

The engineering team created a single umbrella hybrid cloud platform, termed the CIO Hybrid Cloud. It would be a hosting environment for both containerized applications and VM-based applications for a heterogeneous set of architectures. Red Hat OpenShift on IBM Cloud allowed the team to host identical environments on premises and in the cloud. This is shown in Figure 3-4 on page 86.

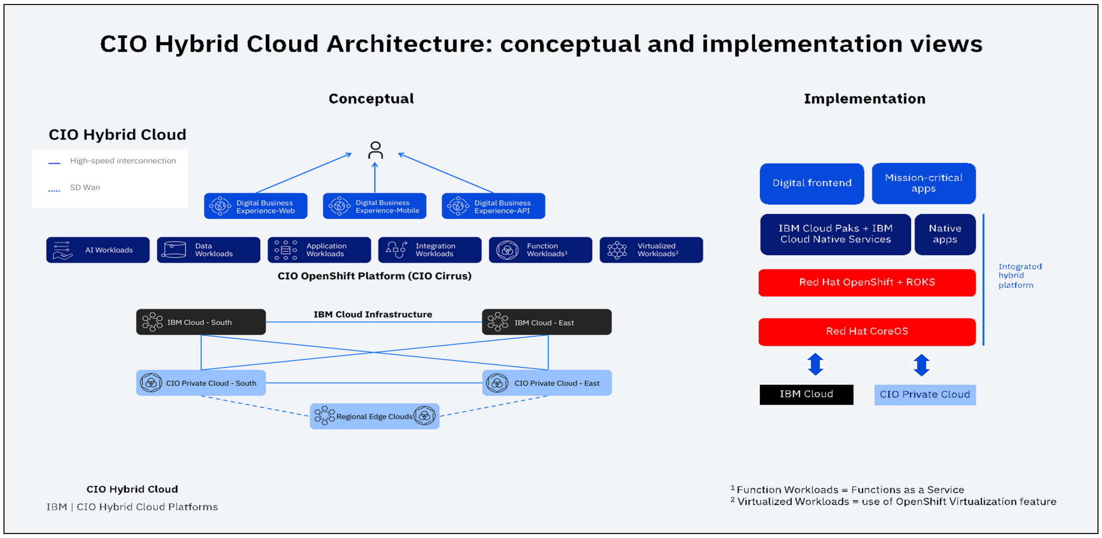

CIO Hybrid Cloud Architecture: conceptual and implementation views
Conceptual
Implementation
CIO Hybrid Cloud
High-speed interconnection
SD Wan
Digital Business Experience-Web
Digital Business Experience-Mobile
Digital Business Experience-API
AI Workloads
Data Workloads
Application Workloads
Integration Workloads
Function Workloads¹
Virtualized Workloads²
CIO OpenShift Platform (CIO Cirrus)
IBM Cloud Infrastructure
IBM Cloud - South
IBM Cloud - East
CIO Private Cloud - South
CIO Private Cloud - East
Regional Edge Clouds
Digital frontend
Mission-critical apps
IBM Cloud Paks + IBM Cloud Native Services
Native apps
Red Hat OpenShift + ROKS
Red Hat CoreOS
Integrated hybrid platform
IBM Cloud
CIO Private Cloud
CIO Hybrid Cloud
IBM | CIO Hybrid Cloud Platforms
¹ Function Workloads = Functions as a Service
² Virtualized Workloads = use of OpenShift Virtualization feature

Figure 3-4 CIO hybrid cloud architecture

“With ever increasing business requirements around security, resiliency and cost optimization, it was important to deliver a hybrid cloud platform that let IBM developers focus on creating enterprise applications for their stakeholders. The CIO Hybrid Cloud platform helps ensure that requirements are met by simply using the platform for hosting. The set of IBM Cloud services integrated to our hybrid cloud platform allows the right balance of application innovation with guardrails in areas such as databases, observability and storage.”

Figure 3-5 shows the hybrid cloud implementation.

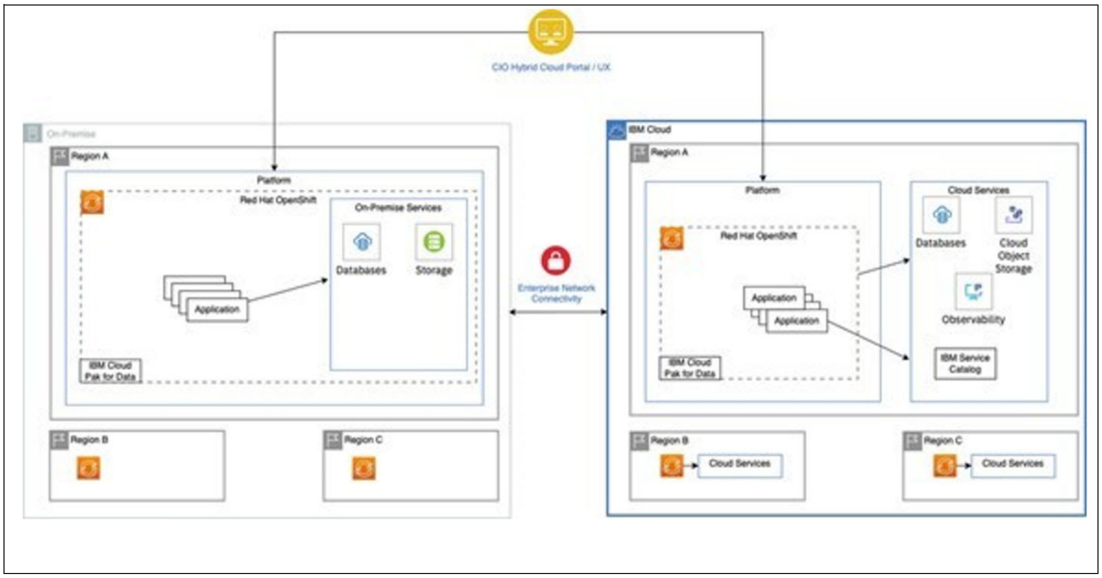


Figure 3-5 Hybrid cloud implementation

The expansion of OpenShift-based Cirrus to Power was part of the IWP initiative (Intelligent Workload Placement), a strategy that will allow OpenShift-deployed apps to run transparently on multiple hardware architectures, including x86, Power or Z. It also accelerates application modernization (from monolithic applications to microservices and other modern application development and deployment techniques).

OpenShift on Power started gaining traction around 2017-2018, with IBM and Red Hat collaborating closely to enable the platform. In 2019, Red Hat announced extended update support (EUS) for OpenShift on Power, allowing clients to stay on a specific version for a longer period with enterprise support.

Towards the end of 2019, IBM launched PowerVS, an Infrastructure-as-a-Service (IaaS) solution, providing simplified automation and accelerated learning for clients. One year later, in 2020 we saw the release of OpenShift 4.6, which included several new capabilities for Power, such as support for compute nodes with up to 512 threads and enhanced storage options. Additionally, Red Hat OpenShift 4.6 became available on IBM Z, IBM Power and x86 platforms simultaneously, reinforcing OpenShift as a platform that is truly “everywhere”.

The IBM CIO Power Virtualization Service delivers end-to-end virtualization services on IBM Power Systems, which drive the maximum level of availability, standardization, and automation to IBM customers. The CIO Power Virtualization Service team manages and supports all on-premise virtualized Power Systems infrastructure across all CIO data centers, providing the strategic standard building blocks to grow to a best in-industry solution:

- On-going management, support and configuration of the global Power Systems infrastructure
  - VIOS, HMCs, PowerVC, Storage, Network VLANs and zone architecture
- Model ourselves after “what's best” in the industry, and leverage commercial IBM Consulting practices where appropriate and educate Application Hosting Service Squads with information they need about these practices.
- Consolidate to a smaller number of management support processes, and standardize tooling and monitoring
- Be proactive in the creation of automated solutions to become more efficient in the support of the global IBM Power environment
- Leverage new and existing technologies to move towards 100% transparent maintenance practice (no outages for planned maintenance) and to reduce the scope, impact and duration of unplanned outages
- Apply IBM Security® Processes to ensure proper health and compliance of the Power Systems infrastructure
- Track and action all security vulnerabilities associated with the Power Systems service
- Collaborate with and keep open communications between the Physical and Automation squads

IBM Power Squad team delivered OpenShift cluster on Power infrastructure integrated with Cirrus in the period of one quarter. It brought to Cirrus the possibility to move some workloads from cloud to private cloud and the modernization of the applications current running on power as AIX and RHEL.

Benefits gained

Red Hat OpenShift on IBM Power optimizes infrastructure costs by reducing the number of servers needed without impacting performance. Figure 3-6 on page 88 show the TCO impact of using IBM Power compared to x86 servers with OpenShift.

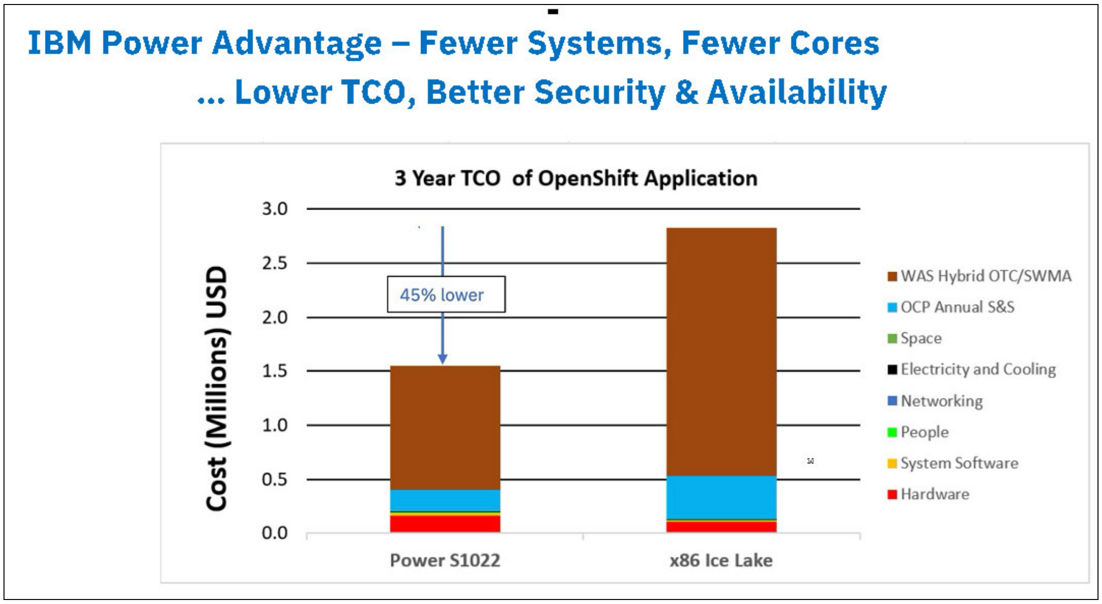


Figure 3-6 TCO Benefits of Power versus x86 for OpenShift

The results were stunning:

- Over 2,000 application components onboarded to CIO Hybrid Cloud platform, mostly custom-built applications, application integrations and data components.
- 55% fewer DevOps hours spent on operations, allowing applications teams to focus on business value.
- 90% cost savings for CIO Hybrid Cloud platform hosting containerized workloads versus legacy hosting.

Application teams are meeting corporate objectives:

- Application changes were taking months, but now changes can be delivered daily. Security patches are delivered rapidly.
- Security and availability were application afterthoughts, but now highly available secure applications are the default.
- Critical application teams struggled with disaster recovery implementation, testing and associated costs. Now disaster recovery support is built into the platform's implementation patterns.

Increasing scale and capabilities of the previous private cloud solution required capital expenditures and operational overhead 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 are used by the platform and by the applications. The cloud provides security-rich resources with low operational overhead and consumption-based pricing. Data and applications are located 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 generated by the applications are stored in a data warehouse and are available for platform and application analysis. Over time some applications will require updates and platform metrics will inform the required changes. Other applications will reach end of life and the impact of retiring an application can be assessed. Application owners have visibility to metrics through a common interface to reason about 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 partners.

The platform can apply AI-based analysis on the data warehouse. Vertical scaling of application hosting resources using the IBM Turbonomic solution is available to the applications. Intelligent workload placement is also possible, allowing applications to migrate for optimal cost and performance on x86, IBM Power, or IBM Z platforms. This provided:

- 90% cost savings – leveraging hybrid cloud containerized workloads
- 55% operations reduction – fewer platform operations resources
- Significant reductions in applications and software licenses

Application success by default

The CIO Hybrid Cloud hosts applications distributed over private and public cloud. The application environment starts with a “batteries included” template with the company business rules built in. A common application runtime environment is reproduced in all data centers that allows applications to run locally and handle regional disruptions. Applications are running on compute and storage that reflects requirements and consuming cloud catalog services with consumption-based pricing.

Moving to a hybrid cloud containerized environment has reduced the hosting expenses dramatically. Traditional virtual machines have been replaced with containerized workloads. Application-specific operational load has been invested in a common platform – allowing teams to focus on the business value while reducing operational expenses.

The IBM CIO hybrid cloud, combined with application modernization and migration will:

- Provide end-to-end observability.
- Consistently operate with speed, scale, security.

The platform is not done; it probably never will be. Compliance requirements will continue to evolve. Application performance demands have spiked with the adoption of AI. It is possible to further optimize the placement of applications in private or public by further analyzing real-time application and cost data.

3.4 Continuous integration/continuous deployment in IBM Cirrus

Continuous integration/continuous deployment (CI/CD) is a set of practices that enable development teams to merge code changes frequently into a central repository and confidently deploy them for users.

As part of our hybrid cloud journey, the IBM CIO organization discovered that a key component of our transformation is safely building and deploying applications at scale. Therefore, we're developing a common organization-wide CI/CD solution built on Red Hat's OpenShift Pipelines. This common pipeline approach allows our organization to build on previous work to scale best practices across the organization. But why are we doing this? I'll lay the foundation for why not just CI/CD, but a common CI/CD solution is beneficial.

3.4.1 What is CI/CD?

CI/CD is a two-step process that removes friction from the development and delivery process using automation. CI is a practice where the development team frequently commits changes into the shared repository and then leverages automation to check that the code still works as expected. This happens by having each commit trigger a build and a series of automated tests to verify the application's behavior and the clean integration of updates. This enables the application team to deploy code frequently, which leads to continuous deployment. CD automates releasing an app to production. Because it is automated with no manual gates, CD relies heavily on well-designed test automation. Once the application passes all tests, a developer's change to an application can go live within minutes

3.4.2 How is CI/CD implemented?

CI/CD automates the process of integrating, releasing, and deploying software while removing roadblocks. This is done with smaller code changes, continuous testing, real-time feedback, faster releases and improved customer satisfaction, and reduced costs:

- Smaller code changes: CI/CD encourages developers to integrate small pieces of code. Smaller code changes are simpler and easier to handle than trying to integrate a large amount of code. This is especially true when a large number of developers are working on the same code base.
- Continuous testing: Building on smaller changes, continuous testing allows these smaller pieces of code to be tested as soon as they are checked into the repository so that developers can see and correct problems quickly.
- Real-time feedback: CI/CD is a great way to get continuous feedback not only from end users but also from the developer team to improve team transparency and accountability. This real-time feedback increases visibility into problems with the team and encourages responsible accountability.
- Faster releases and improved customer satisfaction: CI/CD pipelines provide a means for consistent builds, tests, and deployments, enabling errors to be detected faster. This allows teams to safely release code faster and with new features and bug fixes. Development teams can meet users' needs through regular (even daily!) updates and responding to feedback with rapid, high-quality changes.
- Reduced costs: CI/CD pipelines help catch bugs before they reach production, significantly reducing the costs of addressing them. Deployment automation also reduces the chances of human error affecting deployment.

3.4.3 A security example

In late 2021, a zero-day vulnerability was discovered in the popular logging library Log4j 2. This vulnerability enabled attackers to use a simple string of text to trick a system into requesting and executing malicious code. This vulnerability was rated at the highest possible CVSS severity level and estimated to affect over 90% of enterprise cloud environments.

In addition to the obvious security issues, this incident illustrated that many organizations do not know what components make up their systems or, in some cases, even what systems they are running. It demonstrated the need for organizations to understand what their software is built from and any additional security requirements it demands.

You may be wondering what the Log4j 2 incident has to do with CI/CD. This incident highlighted the need for enterprises to build guide rails into their processes to help them gain better insight into their security positions, something a common CI/CD process can enable.

Security guard rails

Enterprises and chief security officers worldwide have seen recent high-profile vulnerabilities as a wake-up call to rethink security. These new approaches include:

- 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) vulnerability detection
- Dynamic Application Security Testing (DAST) vulnerability detection
- Container image vulnerability detection
- Artifact signing

These new activities may require significant compliance work for development teams. If you understand that CI/CD helps team catch errors early in the process and make security part of the regular development process, CI/CD pipelines are a natural place to start, and a common CI/CD makes it even easier. With a common CI/CD in place, you can add new security tools without affecting existing development processes and developers.

This is the approach we are taking with our common CI/CD journey. By adding security to our CI/CD process, we are introducing guide rails for our developers that help them focus on delivering business value while also protecting the enterprise. Our common approach provides a process that:

- Implements our security policies, including scanning
- Reduces developer response time for issues like Log4J
- Promotes application quality and consistency
- Offloads compliance and ongoing maintenance burdens

In addition, on the CD side, we can provide:

- Production deployment approval auditing
- DAST processes
- Deployment frequency and duration metrics
- Deployment region location information

Economy of scale and friction reduction

Most development teams agree on the value of a CI/CD pipeline; however, they also may face barriers to adoption that may be related to costs, skills, or time. And as more requirements are placed on teams, these barriers can get higher.

These barriers and ever-increasing compliance requirements create friction in the development process that can reduce developer productivity. From the organization's point of view, every team implementing its own CI/CD approach creates several inefficiencies: each team spends time creating and maintaining the pipelines, procuring and managing the infrastructure, and integrating any new tools. Combined, these increase the costs and delays in deploying features.

A common CI/CD provides a way to address these problems. A single team is responsible for creating, maintaining, and standardizing the pipelines; supporting and managing the infrastructure; and integrating new tools. For our development teams, this means:

- Greater ability to focus on business value and not process and infrastructure
- Increased productivity and satisfaction
- Accelerated application modernization

Altogether, this reduces the burden of CI/CD adoption for development teams and friction in the development process. Couple this with the ability to leverage a single infrastructure, and an economy of scale starts to emerge.

Moving to a common CI/CD approach helps our organization collect and analyze data from the pipelines. This enables greater insight into an application's source code all the way to deployment from a common set of tools pushing into a common data store. We've developed of a data lake to bring the data together and a developer experience portal to provide a single pane of glass that makes data easy to understand.

For the IBM CIO organization, our approach allows us to:

- Better understand our open source usage
- Better understand application quality
- Gain insights into our application runtimes and languages
- Reduce response time when security issues arise, such as Log4J

CI/CD is a set of practices that enable teams to safely build and deploy applications. Introducing a common CI/CD solution provides a means to scale the benefits of CI/CD across an organization while reducing the overall adoption effort. With such an offering in place, we can add security compliance activities to the pipelines without overloading teams. This results in not only a safer, more knowledgeable organization, but also happier and more productive developers.

3.5 Computer Systems Integration Ltd (CSI Ltd)

CSI is a managed service and integration provider based in the UK and has a presence in the US as Tectrade. CSI is a Platinum IBM partner dealing with IBM products for the last 40 years focusing on IBM Power Systems, HPC, Storage and High-End Vendor Integration Solution.

Figure 3-7 on page 93 shows CSI Innovation Lab which is the result of long-term collaboration between CSI & IBM. Initially targeted to demonstrate IBM’s vision of Enterprise Hybrid Cloud and Modernized solutions, the project has evolved and grown each year with pre-GA, Early Adoption Programs and First of a Kind Hybrid Cloud, Modern and AI solutions. CSI demonstrate how Enterprise IBM Power integrates seamlessly with other Hybrid Cloud vendor solutions. This solution has been implemented in a one-of-a-kind DEMO environment call CSI Innovation Lab.

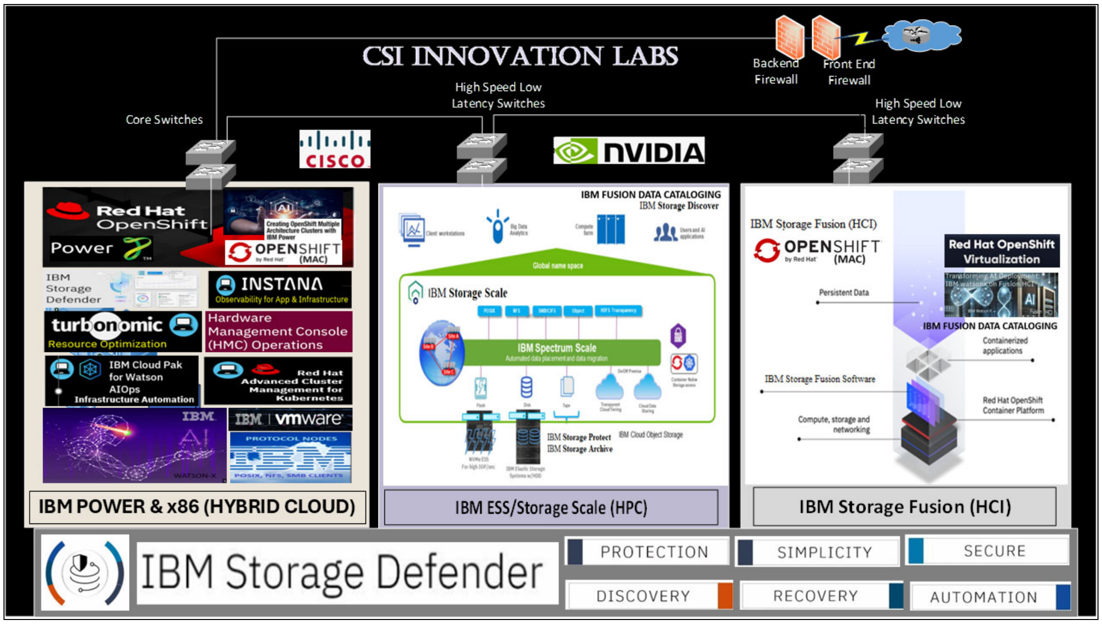

CSI INNOVATION LABS
Backend Firewall
Front End Firewall
High Speed Low Latency Switches
High Speed Low Latency Switches
Core Switches
CISCO
NVIDIA
Red Hat OpenShift
Power
Creating OpenShift Multiple Architecture Clusters with IBM Power
OPENSHIFT by Red Hat (MAC)
IBM Storage Defender
INSTANA
Observability for App & Infrastructure
turbonomic
Resource Optimization
Hardware Management Console (HMC) Operations
IBM Cloud Pak for Watson AIOps Infrastructure Automation
Red Hat Advanced Cluster Management for Kubernetes
IBM AI
WATSON-X
IBM | vmware
PROTOCOL NODES
POSIX, NFS, SMB CLIENTS
IBM POWER & x86 (HYBRID CLOUD)
IBM FUSION DATA CATALOGING
IBM Storage Discover
Client workstations
Big Data Analytics
Compute farm
Users and AI applications
Global name space
IBM Storage Scale
POSIX
NFS
SMB/CIFS
Object
HDFS Transparency
IBM Spectrum Scale
Automated data placement and data migration
Site A
Site B
Site C
On/Off Premise
Container Native Storage access
Flash
Disk
Tape
Transparent Cloud Tiering
Cloud Data Sharing
NVMe ESS For high IOP/sec
IBM Elastic Storage Systems w/HDD
IBM Storage Protect
IBM Storage Archive
IBM Cloud Object Storage
IBM ESS/Storage Scale (HPC)
IBM Storage Fusion (HCI)
OPENSHIFT by Red Hat (MAC)
Red Hat OpenShift Virtualization
Transforming AI Deployment: IBM watsonx on Fusion HCI
Persistent Data
IBM FUSION DATA CATALOGING
Containerized applications
IBM Storage Fusion Software
Red Hat OpenShift Container Platform
Compute, storage and networking
IBM Storage Fusion (HCI)
IBM Storage Defender
PROTECTION
SIMPLICITY
SECURE
DISCOVERY
RECOVERY
AUTOMATION

Figure 3-7 Diagram of CSI Innovation Lab environment

The solution showcases IBM’s vast portfolio of technologies with major focus on leveraging AI to meet practical use cases along with other vendors, working seamlessly to provide both “art of the possible” and production ready solutions. The above solution breaks down into three major categories.

- IBM Power 10 Series & AI, x86 Platforms together, called HYBRID CLOUD which is the CORE of the solution.
- IBM Storage Scale with Storage Scale System family of storage solutions for HPC workloads
- IBM Fusion HCI for AI, OpenShift Virtualization, and containers with built-in backup and restore and Fusion Data Cataloging.

3.5.1 IBM Power and x86 Hybrid Cloud components

The Hybrid Cloud components allows a customer to explore the integration of IBM Power and x86 architectures in a hybrid cloud environment, leveraging industry-leading technologies such as:

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

Figure 3-8 shows the hybrid cloud components.

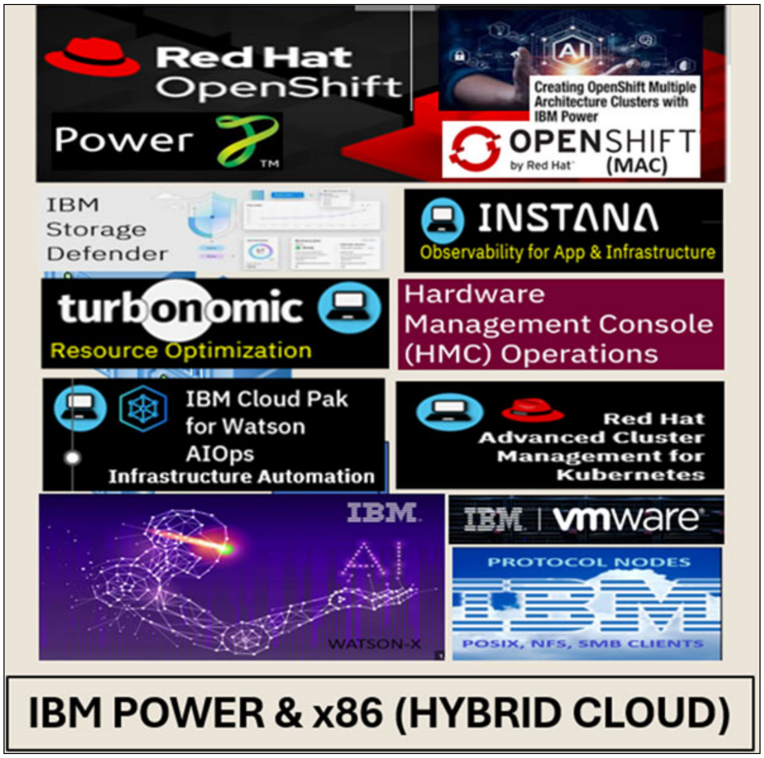

Red Hat
OpenShift
Power
AI
Creating OpenShift Multiple Architecture Clusters with IBM Power
OPENSHIFT
by Red Hat
(MAC)
IBM Storage Defender
INSTANA
Observability for App & Infrastructure
turbonomic
Resource Optimization
Hardware Management Console (HMC) Operations
IBM Cloud Pak for Watson AIOps
Infrastructure Automation
Red Hat Advanced Cluster Management for Kubernetes
IBM
AI
WATSON-X
IBM | vmware
PROTOCOL NODES
IBM
POSIX, NFS, SMB CLIENTS
IBM POWER & x86 (HYBRID CLOUD)

Figure 3-8 Hybrid cloud components f

The hybrid cloud provides a workspace that allow clients to gain insights into optimizing infrastructure for modern workloads, improving observability, and enhancing resource management.

Key technologies covered

The Hybrid Cloud provides a view into these key technologies:

- Red Hat OpenShift & Kubernetes

  Deploying multi-architecture compute OpenShift clusters, including IBM Power and x86 Worker Nodes.

- Red Hat Advanced Cluster Management:

  Provides a single interface to manage and govern multiple Hybrid Cloud OpenShift Clusters.

- IBM Cloud Pak for Watson AIOps Infrastructure Automation:

  AI-driven infrastructure automation and anomaly detection.

- Turbonomic Resource Optimization

  Automatically optimizing workload management and cost efficiencies.

- Instana Observability

  Real-time observation and monitoring of applications and infrastructure.

- Hardware Management Console (HMC)

  Best practices for managing IBM Power systems.

- IBM and VMware Integration

  Streamlining hybrid cloud operations with seamless compatibility.

- IBM watsonx.ai®

  Unlocking AI-driven insights and automation in cloud environments.

- IBM Storage Defender

  Enhancing security and data protection in a hybrid cloud environment.

3.5.2 IBM Storage Scale and IBM Storage Fusion

As part of the lab, we have also integrated the Hybrid cloud with IBM ESS/Storage Scale and IBM Storage Fusion (highlighting their role in modern AI, analytics, and enterprise workloads). It focuses on IBM Spectrum Scale for scalable data management and IBM Storage Fusion for containerized environments with Red Hat OpenShift.

Figure 3-9 shows the storage components connected to the hybrid cloud.

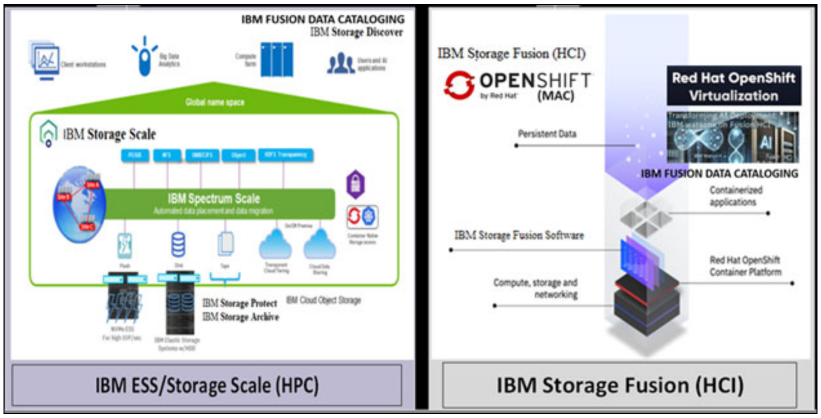

IBM FUSION DATA CATALOGING
IBM Storage Discover
Client workstations
Big Data Analytics
Compute farm
Users and AI applications
Global name space
IBM Storage Scale
IBM Spectrum Scale
Automated data placement and data migration
IBM Storage Protect
IBM Storage Archive
IBM Cloud Object Storage
IBM ESS/Storage Scale (HPC)
IBM Storage Fusion (HCI)
OPENSHIFT
by Red Hat
(MAC)
Red Hat OpenShift Virtualization
AI
Persistent Data
IBM FUSION DATA CATALOGING
Containerized applications
IBM Storage Fusion Software
Red Hat OpenShift Container Platform
Compute, storage and networking
IBM Storage Fusion (HCI)

Figure 3-9 Storage components in the hybrid cloud

When used in combination with Red Hat OpenShift Multi-Architecture Compute capability, this enables seamless movement of Applications (Pods) between cluster nodes on x86 and on IBM Power systems. Instana provides real time observation and monitoring, integrating with Turbonomic to automate management and provide cost efficiency. This capability is unique to Red Hat OpenShift Multi-Architecture Compute with both IBM Power and x86 Worker Nodes.

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

- IBM ESS/Storage Scale (HPC)

  A high-performance, scalable storage solution built on IBM Storage Scale (previously know as Spectrum Scale) for automated data placement and migration.

- IBM Storage Fusion (HCI)

  A hyper-converged infrastructure solution leveraging Red Hat OpenShift and IBM Storage Fusion Software for persistent data storage, containerized applications, and AI workloads.

IBM Storage Scale

IBM Storage Scale provides a global namespace, automated data management, and cloud integration. to provide a highly scalable enterprise level storage solution. IBM Storage Scale Systems (SSS) – previously called Elastic Storage Systems (ESS) provide a purpose-built hardware implementation of IBM Storage Scale (even earlier named General Parallel File System – GPFS) ensures high IOPS and flexible storage with NVMe and HDD options with policy driven data placement. For more information on IBM Storage Scale see https://www.ibm.com/products/storage-scale.

IBM Storage Fusion

IBM Storage Fusion is a software-defined, AI-driven infrastructure for Red Hat OpenShift environments with built-in Fusion Data Cataloging (earlier called Spectrum Discover) which includes backup and restore of containers and virtual machines. It is delivered as software only solution or as part of an appliance-based solution.

Red Hat OpenShift Virtualization

Red Hat OpenShift Virtualization enhances AI deployment through containerized storage solutions. This also provides a Platform for VM Migrations from high-cost platforms.

Consider the following use cases

- AI/ML Workloads with IBM watsonx on Storage Fusion HCI
- High-performance computing (HPC) with
- SSS and Storage Scale
- Cloud-native storage for containerized applications
- Data protection and archiving with IBM Storage Protect and Archive.

4

Services and Consulting Option

Modernization projects are often complex and include a high level of risk. Engaging services and consulting for infrastructure modernization offers businesses a strategic advantage by providing access to specialized expertise and experience. Modernization projects often involve intricate technologies and complex implementations, where consultants can mitigate risks and minimize costly errors through their in-depth knowledge and up-to-date understanding of industry best practices. They assist in developing comprehensive modernization strategies aligned with specific business goals, assess current infrastructure, and create detailed implementation roadmaps. Furthermore, consultants play a crucial role in risk mitigation and cost optimization, identifying potential disruptions and recommending efficient solutions to avoid unnecessary expenses.

By leveraging their partnerships with technology vendors, businesses gain access to a wider range of resources and solutions, ensuring the selection of appropriate technologies. Ultimately, outsourcing infrastructure modernization allows businesses to focus on their core competencies, freeing up internal resources and maintaining productivity while consultants manage the technical aspects of the project, ensuring a smooth and efficient transformation.

This chapter provides a look at some of the services that IBM and IBM business partners offer clients that assist them in their modernization journey. The following topics are covered:

© Copyright IBM Corp. 2025.

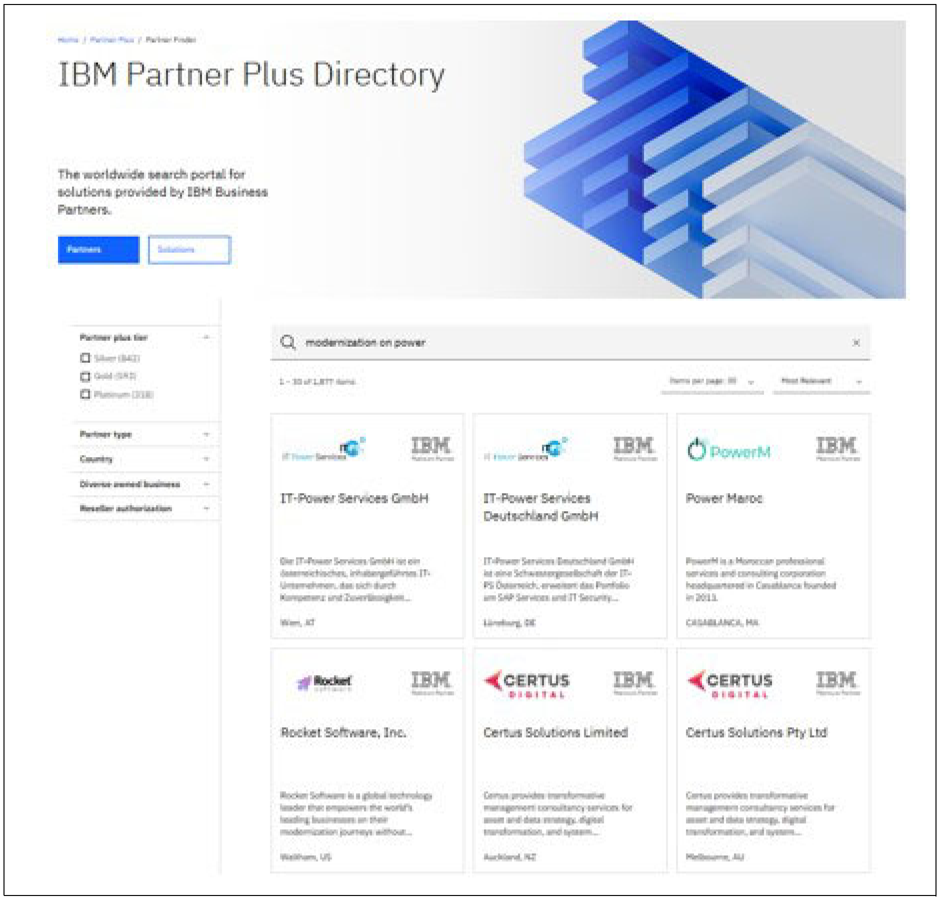
4.1 Business partners

IBM has a comprehensive ecosystem of Business Partners who work with them to deliver solutions and services to clients across various industries. These partnerships are crucial for extending IBM's reach, providing specialized expertise, and creating innovative offerings.

These collaborations encompass a wide spectrum of organizations, including solutions providers, system integrators, managed service providers, independent software vendors, and consultancies, all working together to leverage IBM's cutting-edge technologies.

4.1.1 IBM Partner Plus Directory

Whether you are a business partner or just a company requiring support, you can search the IBM Partner Plus® Directory to find IBM Business Partners specializing in services and modernization methods on IBM Power. A quick search at the time of writing this document produces 1,877 business partners around the world matching the phrase “modernization on power”.



Home / Partner Plus / Partner Finder

IBM Partner Plus Directory

The worldwide search portal for solutions provided by IBM Business Partners.

modernization on power

1 - 30 of 1,877 items

IT-Power Services GmbH

Die IT-Power Services GmbH ist ein österreichisches, inhabergeführtes IT-Unternehmen, das sich durch Kompetenz und Zuverlässigkeit...

Wien, AT

IT-Power Services Deutschland GmbH

IT-Power Services Deutschland GmbH ist eine Schwestergesellschaft der IT-PS Österreich, erweitert das Portfolio um SAP Services und IT Security...

Lüneburg, DE

Power Maroc

PowerM is a Moroccan professional services and consulting corporation headquartered in Casablanca founded in 2011.

CASABLANCA, MA

Rocket Software, Inc.

Rocket Software is a global technology leader that empowers the world's leading businesses on their modernization journeys without...

Waltham, US

Certus Solutions Limited

Certus provides transformative management consultancy services for asset and data strategy, digital transformation, and system...

Auckland, NZ

Certus Solutions Pty Ltd

Certus provides transformative management consultancy services for asset and data strategy, digital transformation, and system...

Melbourne, AU

Figure 4-1

IBM business partners provide IBM certified solutions backed by premium IBM support around the world. This means that the technology and solution can be explained and delivered in a language and a manner that you the customer understands, and are comfortable with.

For example, IBM works with their platinum business partner Saudi Business Machines in the Middle east to provide exceptional solutions for Modernization on IBM Power Systems. Where SBM provides the infrastructure and IBM works with ISVs and the customer to provide software solutions with IBM Cloud Paks for Integration.

In a modernization scenario on IBM Power, upgrading the infrastructure is usually always the starting point, to provide the compute resources and virtualization layer to be able modernize.Local IBM business partners provide that cohesion required to move the project forward successfully.

Note: Search the Partner Plus Directory for a certified IBM Business Partner.
https://www.ibm.com/partnerplus/directory/companies

Figure 4-2 shows an example of a modernization project including IBM Power with IBM Cloud Pak for Integration.

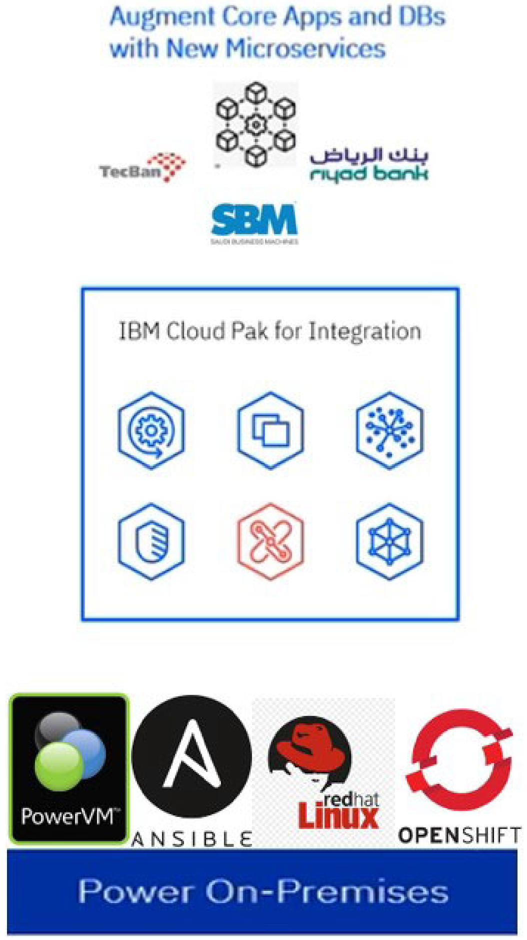


Figure 4-2 Example of modernization project components

4.2 IBM Technology Expert Labs

IBM Technology Expert Labs is a professional services organization powered by an experienced team of product experts. This knowledgeable team brings deep technical expertise across software and infrastructure. Their skills include IBM data and AI, automation, sustainability, security, software-defined networking, IBM Power, IBM Storage, IBM Z and LinuxONE, IBM Z software, IBM GDPS and IBM Cloud.

IBM Technology Expert Labs mission is to accelerate adoption (deployment, consumption, expansion) of the IBM Hybrid Cloud and AI strategy with experts, practices, and frictionless digital experiences for continuous client business outcomes.

Our Technology Expert Labs Infrastructure consultants perform services for clients online or on-site, offering deep technical expertise, valuable tools and successful methodologies. Technology Expert Labs has a global presence and can deploy experienced consultants around the world. Our experts have experience working with large global customers across domains including Financial, Insurance, Healthcare and Retail on Modernization with IBM Power.

Why work with us:

1. A proven methodology - We use proven methodologies, practices and patterns to help you achieve better business outcomes.
2. Specialized industry experts - We develop complex solutions and minimize implementation risks.
3. An extension of development - Our ties to development mean first-in-line access to product insights, features and solutions.
4. Accelerate successful adoption of your IBM infrastructure - We work with you to achieve value faster from IBM products through various offerings that span across project lifecycle. These offerings include assessment, planning, implementation, migration, upgrade and post-implementation advisory services.

Some of the popular modernization use cases on IBM Power that our consultants have helped with include:

- Planning and deployment with Red Hat OpenShift on IBM Power for IBM Power10 Private Cloud Rack for Db2 Warehouse
- Plan and implement Red Hat OpenShift on IBM Power to benefit from Application Modernization for banking applications
- Modernize IBM i with Merlin on Red Hat OpenShift for CI/CD setup
- Migrate workloads to IBM Power Virtual Server on IBM Cloud for AIX, IBM i and SAP HANA
- Leverage HashiCorp Terraform and Red Hat Ansible based automation for provisioning and hardening day-2 operations such as patch management and LPAR management
- Build IBM PowerVC to provide IBM Private Cloud environment for IBM Power systems to reduce provisioning time and optimize VM management
- Build Shared Utility Capacity for cloud like consumption model

Refer to IBM Technology Expert Labs for Systems Standard Offerings for pre-scoped and pre-priced Standard Offerings for faster solution implementation in your organization.

Our advisory subscription-based Expertise Connect offering is available to augment your team with deep technical expertise with flexible options to achieve your goals from project-based assistance to on-demand expertise related to ongoing operations.

You can Partner with IBM Technology Expert Labs and leverage our services in migrating your on-premises IBM Power Servers to IBM Power Virtual Server in IBM Cloud

Visit us at https://www.ibm.com/products/expertlabs/infrastructure to get more information or reach out to us at systems-expert-labs@ibm.com.

For IBMers and Business Partners refer to ibm.biz/systemsexpertlabsofferings-sales

4.3 Client Engineering

Client Engineering is an investment by IBM to jointly innovate and rapidly prove solutions to your business opportunities by leveraging IBM hybrid cloud and AI technologies and our mission is shown in Figure 4-3.

Client Engineering | IBM

IBM Client Engineering experts co-create with you to solve business problems using open source and IBM technologies.

Figure 4-3 Client engineering mission

1. What do we offer to clients?

   A no-cost IBM multi-disciplinary team and expertise to jointly innovate and rapidly prove solutions to client's business needs, leveraging IBM technologies.

2. What value do clients get?

   Confidence in a technical solution to your business needs and accelerating time to value.

3. How do we do it?

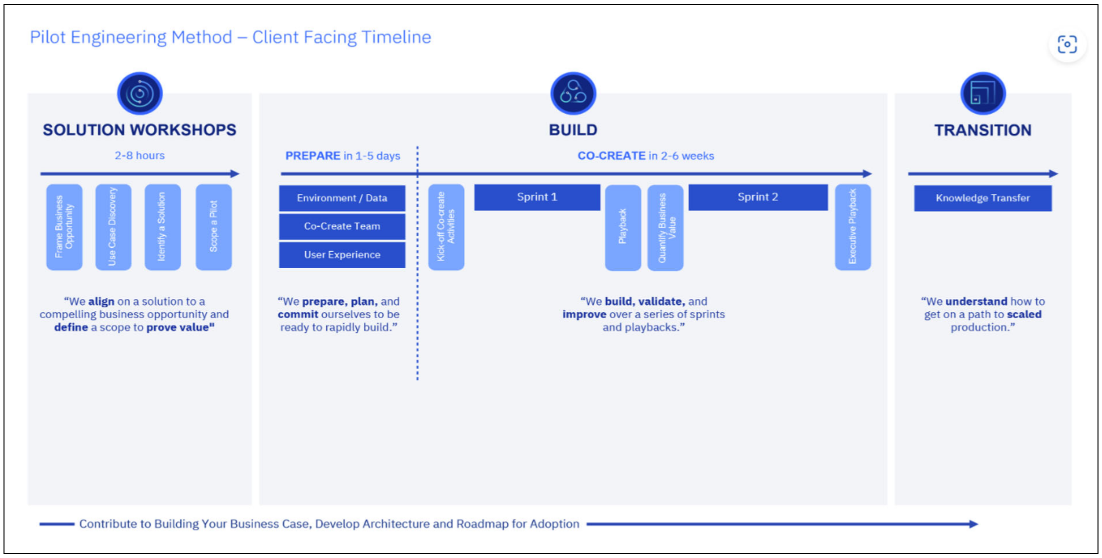
   We innovate, iterate, and prove using IBM's user-centric Pilot Engineering Method through our accelerators. We deploy over 1,700 experts in multi-disciplinary squads to co-create with clients.

What is a Pilot?

A pilot is a rapid co-creation build that proves the value of IBM technologies to deliver on a client's desired business outcomes. The pilot uses a user-centric approach to identify high-impact use cases, define the pilot scope, contribute to a business case, and build a solution that meets the client's pilot requirements. IBM and the client working closely together builds confidence in the solution, fosters a collaborative partnership for long-term success, and provides a foundation for further development and scale.

Pilot Engineering Method

Figure 4-4 on page 102 shows our Pilot Engineering Method which we use in our client engagement. We start with solution workshops to define the requirements and align our solution with your business requirements. We then work with you to build the solution, ultimately getting you ready to transition the solution to your environment.


Pilot Engineering Method – Client Facing Timeline

"We **align** on a solution to a compelling business opportunity and **define** a scope to **prove value**"

"We **prepare, plan,** and **commit** ourselves to be ready to rapidly build."

"We **build, validate,** and **improve** over a series of sprints and playbacks."

"We **understand** how to get on a path to **scaled** production."

Contribute to Building Your Business Case, Develop Architecture and Roadmap for Adoption

Figure 4-4 Pilot engineering method

Skills and capabilities

IBM Client Engineering EMEA has teams of SMEs that have a long experience with IBM Power Systems and therefore can conduct such pilots in a Power Infrastructure context optimizing the experience for the Client.

IBM Client Engineering proposes pilots around technologies that matter to our Power Clients such as:

- Power & AI
- Move to the Cloud with PowerVS
- Automation with Red Hat Ansible, PowerVC, HashiCorp Terraform
- Code Modernization with Watson Code Assistant
- Application Modernization with Red Hat OpenShift containerizations

IBM Client Engineering EMEA also works closely with TechZone to provide the "custom environment" required for any pilot of an IBM SW Solution that must also meet specific infrastructure, client standards and/or non-functional requirements.

How to engage

Engage with Client Engineering through your IBM or Business Partner Sales representatives.

4.4 IBM Consulting

IBM Consulting, formerly known as IBM Global Business Services®, is the professional services and consulting arm of IBM. It serves a wide range of clients, including companies, government organizations, non-profits, and NGOs, offering expertise to advise, design, build, and operate business innovation.

IBM has a significant global footprint with offices and consultants located in numerous countries worldwide. Its headquarters is in Armonk, New York, and it has major offices in locations like Yorktown Heights and San Francisco in the US, Hursley in the UK, and

Bangalore in India, among many others. This global network allows them to serve clients across different geographies and industries.

Today IBM Consulting provides assets and productivity tools that are focused on driving productivity. These can be broadly categorized as:

1. Business Value Assets

   Business value assets are assets such as

   - Software products or platforms that are typically left behind after an engagement. These are typically market priced when released in Asset Catalog. Subscription and/or perpetual license models are available.
   - New Business Value Assets & significant enhancements to Traditional Assets to be onboarded to Advantage.

2. Productivity based assets and assistants

   These are software that drives the productivity of IBM Consultants. Client may or may not use with IBM Consultant.

Use cases

Below are some of standard use cases into Software Development Life Cycle of any project

- Accelerate legacy code business rule extraction
- Optimize the ongoing development and modernization activities.
- Java to microservice development assistant using watsonx code assistant
- Provide 30-40% productivity increase in development phase using GenAI assets and assistants
- Generate pseudocode, framework to framework conversion
- Generate a test case scenario, performance test plan and test data preparation

Additionally IBM Consulting can assist you in:

- Test Case and Automation Creation

  Generate detailed test cases from project requirements and automated testing scripts compatible with various tools and languages

- Mainframe and Legacy System Support

  Generate test cases for legacy systems

- Data Structuring and Code Conversion

  Transform unstructured test documentation into organized formats and create varied test data

- Performance Testing and Reporting

  Develop comprehensive test plans and reports for early issue detection and system optimization

- AI-Powered Quality Checks

  Compare test results with expected outcomes to ensure testing accuracy and reliability, and reducing the risk of production defects

IBM Consulting acts as a partner for organizations looking to navigate complex business challenges and leverage technology for innovation and growth, drawing upon IBM's vast technological resources and deep industry expertise.

5

Modernizing the Management of IBM Power Servers

Modernizing management capabilities for IBM Power servers is essential to keep up with the demands of a digital-first world, where businesses need to operate with agility, scalability, and security. User interfaces and dashboards play a critical role in effective management. IBM has created user-friendly interfaces, such as the IBM Power Systems Hardware Management Console (HMC) and the IBM Navigator for i, but as the number of systems to be managed increase, the more important it is to integrate automation and even AI driven tools to deploy applications at scale across the enterprise. The key is not just upgrading the hardware but also evolving the management and monitoring tools to support modern workloads efficiently.

Using tools like PowerVC (IBM's virtualization management product based on OpenStack), Ansible, and Terraform provide automation capabilities for management of IBM Power servers. Automation simplifies the management of your systems as the number of systems grows. Automation provides the ability to quickly add new servers into your environment while enforcing standards and ensuring that the resulting systems are secure and efficient.

These management advances are especially important as your invariant expands to hybrid environments including cloud offerings such as IBM Power Virtual Server (where you can run workloads on IBM Power servers in the IBM Cloud) and when you utilize IBM Power Private Cloud with Shared Utility Capacity to optimize your costs with shared resources and a pay-per-use by-the-minute economics model for compute capacity in Power Enterprise Pools 2.0.

The following topics are discussed in this chapter:

- 5.1, "HMC simplification" on page 106
- 5.3, "Power Virtualization Center overview" on page 111
- 5.4, "Ansible OpenStack Modules" on page 128
- 5.6, "Power Enterprise Pools & Cloud Management Console" on page 162
- 5.7, "PowerVM Networking Concepts" on page 163

© Copyright IBM Corp. 2025.

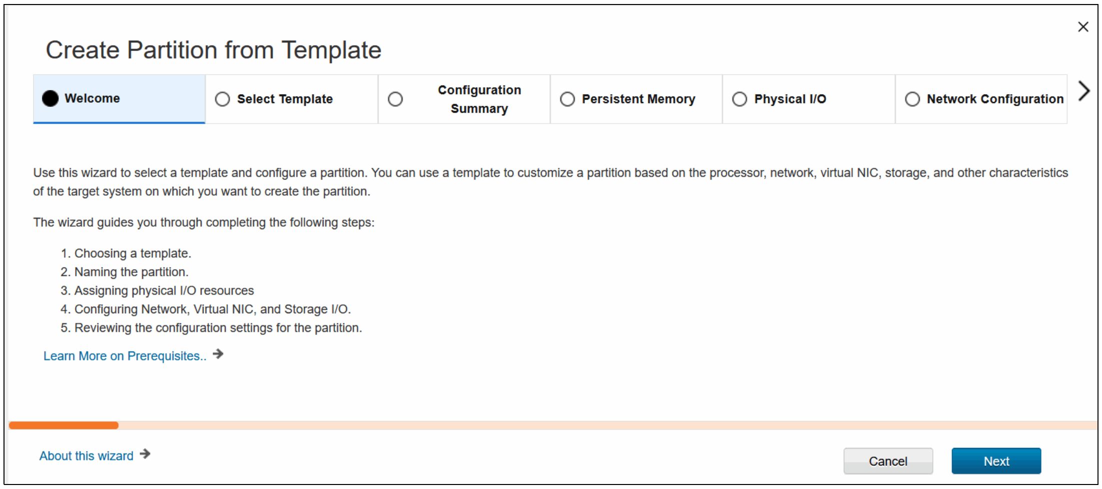
5.1 HMC simplification

Hardware Management Console (HMC) is an appliance for planning, deploying and managing IBM Power Systems. It can be used to create and modify logical partitions, including dynamically adding and removing resources for a running partition.

Over last few years IBM development constantly improves the HMC interface by adding features which help with simplification of operations and automation. Many functions which previously require manual configuration in Virtual I/O Server or HMC CLI can be achieved from the GUI.

5.1.1 System and partition templates

System and partition templates contain details for the system or partition resources, such as number of processors, memory, physical adapters, virtual networks, and storage configuration. A user can quickly deploy servers or create an LPAR from the quick-start templates that are available in the template library or from own user-defined templates stored in the Hardware Management Console as shown in Figure 5-1.



Use this wizard to select a template and configure a partition. You can use a template to customize a partition based on the processor, network, virtual NIC, storage, and other characteristics of the target system on which you want to create the partition.

The wizard guides you through completing the following steps:

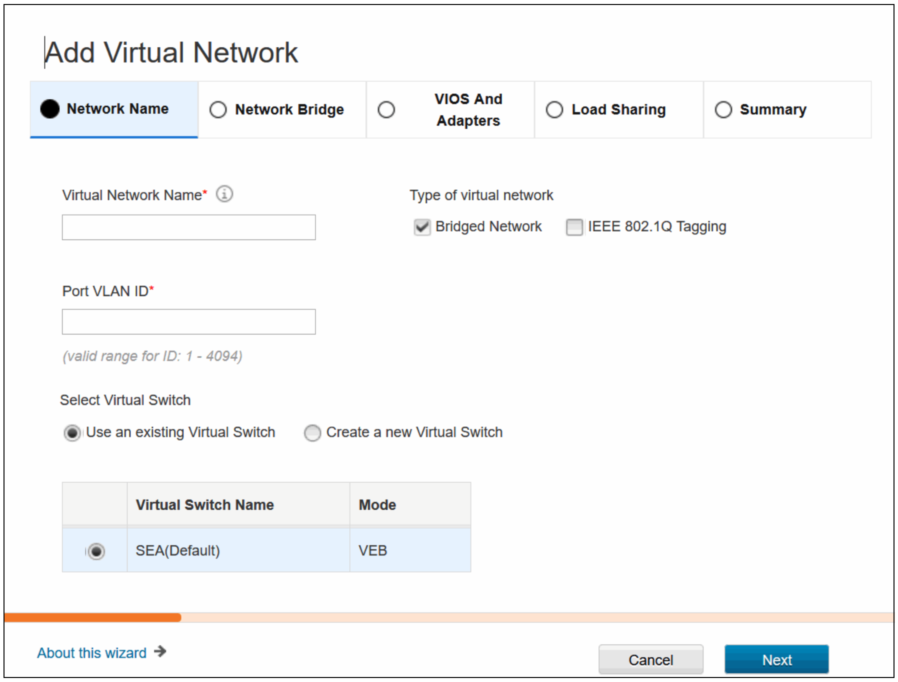
1. Choosing a template.
2. Naming the partition.
3. Assigning physical I/O resources
4. Configuring Network, Virtual NIC, and Storage I/O.
5. Reviewing the configuration settings for the partition.

Figure 5-1 Partition template wizard

5.1.2 Shared Ethernet Adapter simplification

Shared Ethernet Adapter (SEA) is a technology that bridge internal network traffic to a physical network adapter. The Shared Ethernet Adapter eliminates the need for each client logical partition to have a dedicated physical adapter to connect to the external network.

The HMC provides a functionality which allows to create the SEA from the graphical interface without using VIOS command line interface (CLI). The interface provides wizard which creates the SEA in different configurations, and modify features such as, Failover, VLAN tagging, Load Sharing or EtherChannel. With the GUI interface an user can easy adjust VLANs in the existing configuration or modify the parameters without touching CLI. This is shown in Figure 5-2 on page 107.



Add Virtual Network

● Network Name ○ Network Bridge ○ VIOS And Adapters ○ Load Sharing ○ Summary

Virtual Network Name*

Type of virtual network

☑ Bridged Network ☐ IEEE 802.1Q Tagging

Port VLAN ID*

(valid range for ID: 1 - 4094)

Select Virtual Switch

● Use an existing Virtual Switch ○ Create a new Virtual Switch

	Virtual Switch Name	Mode
●	SEA(Default)	VEB

Figure 5-2 Shared Ethernet Adapter creation wizard

5.1.3 Hardware Management Console update and upgrade

It is important to keep the Hardware Management Console firmware on the supported release in order to run entire IBM Power System infrastructure safe and secure. The latest HMC firmware release bring enactments which simplify updating and upgrading process.

HMC update

The HMC update process can fetch all the prerequisites along with the selected update and install in required order. All necessary fixes will be automatically downloaded from IBM website.

HMC upgrade

The HMC upgrade process has been enhanced to automate and simplify the process of upgrading Hardware Management Console. The process includes following steps:

- Save the HMC upgrade data
- Download upgrade files
- Restart and upgrade the HMC

5.2 Virtual I/O Server management improvements

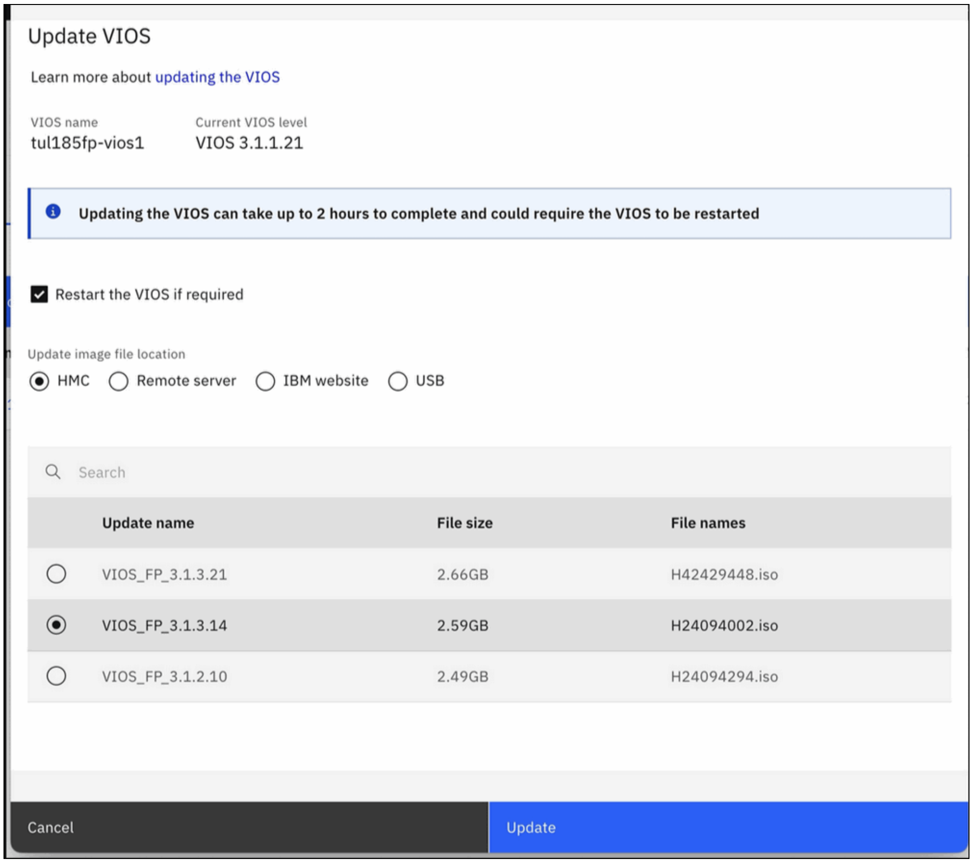
The Virtual I/O Server (VIOS) plays a critical role in PowerVM virtualization, allowing physical I/O adapters to be virtualized and shared across multiple virtual machines (VMs) on a server. VIOS operates as a Logical Partition (LPAR) running a specialized version of the operating system designed for efficient resource sharing. To ensure redundant access to shared adapters, best practices recommend running two VIOS LPARs on a server. VIOS is maintained and updated by IBM, with periodic releases that introduce new features and address known issues. It is the responsibility of the client to keep their VIOS updated to ensure security and minimize downtime.

Recent VIOS releases have simplified maintenance by integrating upgrade functions into the HMC GUI, automating the process to reduce time and minimize the risk of errors. This section describes the new functionality.

5.2.1 VIOS update

The HMC interface introduces function to update the VIOS using HMC interface. VIOS update is the process where new service packs within the same technology level will be installed.

Previously, it was necessary to update the VIOS from the VIOS CLI. As a part of PowerVM simplification, VIOS update images can be imported from a remote NFS or sFTP server, attached in a USB device, or directly downloaded by the HMC from IBM website and installed. The entire process is completely automated and managed from the HMC GUI only. This is shown in Figure 5-3.


Update VIOS

Learn more about updating the VIOS

VIOS name: tul185fp-vios1

Current VIOS level: VIOS 3.1.1.21

Updating the VIOS can take up to 2 hours to complete and could require the VIOS to be restarted

☑ Restart the VIOS if required

Update image file location

● HMC ○ Remote server ○ IBM website ○ USB

Search

	Update name	File size	File names
○	VIOS_FP_3.1.3.21	2.66GB	H42429448.iso
●	VIOS_FP_3.1.3.14	2.59GB	H24094002.iso
○	VIOS_FP_3.1.2.10	2.49GB	H24094294.iso

Figure 5-3 VIOS update wizard

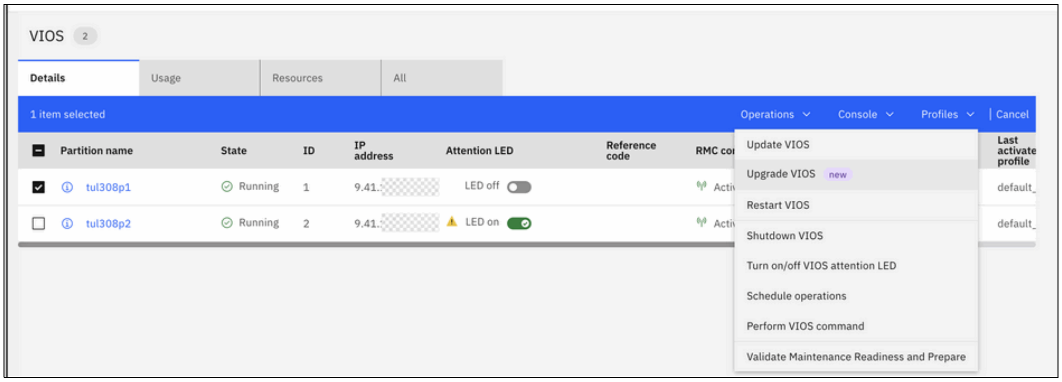
To learn more about the functionality, visit this IBM blog.

5.2.2 VIOS upgrade

VIOS upgrade is the process which need to be used across different Technology Levels (such as 3.0.x to 4.0.x). Previously, it was necessary to use viosupgrade tool in the VIOS CLI. With

PowerVM simplification the entire process can be managed from the HMC GUI. The wizard provides takes user through the entire process and manages all necessary components.

This is shown in Figure 5-4.



☐	Partition name	State	ID	IP address	Attention LED	Reference code	RMC co	Last activate profile
☑	tul308p1	Running	1	9.41.[redacted]	LED off		Acti	default_
☐	tul308p2	Running	2	9.41.[redacted]	LED on		Acti	default_

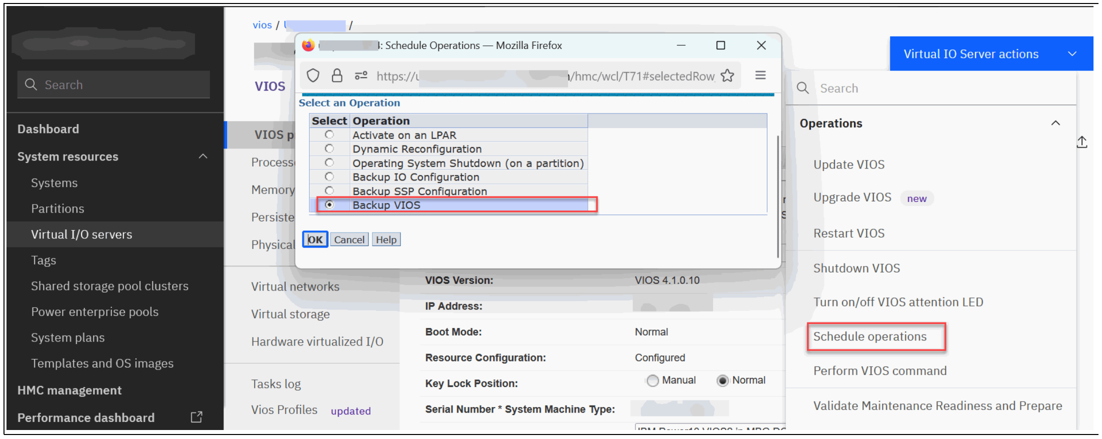
Figure 5-4   VIOS upgrade wizard

To learn more about the functionality, visit this IBM blog.

5.2.3 Automated Virtual I/O Server backups

Backing up the Virtual I/O Server can be fully automated by the HMC interface. PowerVM simplification brings functionality to schedule, and perform VIOS backups. The backup images can be stored in the HMC repository where they can be immediately used for potential restore operation or can be off-loaded to the remote location.

The HMC interface can schedule the entire VIOS backup image (**`backupios`** command) or backup only VIOS IO configuration (**`viosbr`** command) from the GUI. This is shown in Figure 5-5.



Select an Operation

Select	Operation
○	Activate on an LPAR
○	Dynamic Reconfiguration
○	Operating System Shutdown (on a partition)
○	Backup IO Configuration
○	Backup SSP Configuration
●	Backup VIOS

VIOS Version:	VIOS 4.1.0.10
IP Address:	[redacted]
Boot Mode:	Normal
Resource Configuration:	Configured
Key Lock Position:	○ Manual ● Normal
Serial Number * System Machine Type:	[redacted]

Figure 5-5   HMC schedule operation - creating VIOS backup

5.2.4 VIOS restore

The I/O configuration of the Virtual I/O Server can be restored with the HMC GUI but restoring a Virtual I/O Server backup is not supported at this time. It can be achieved from the HMC CLI with **`installios`** command.

To learn more about the functionality see Capturing, Importing, Exporting, and Restoring VIOS backups on the HMC.

5.2.5 Virtual I/O Server validate maintenance readiness

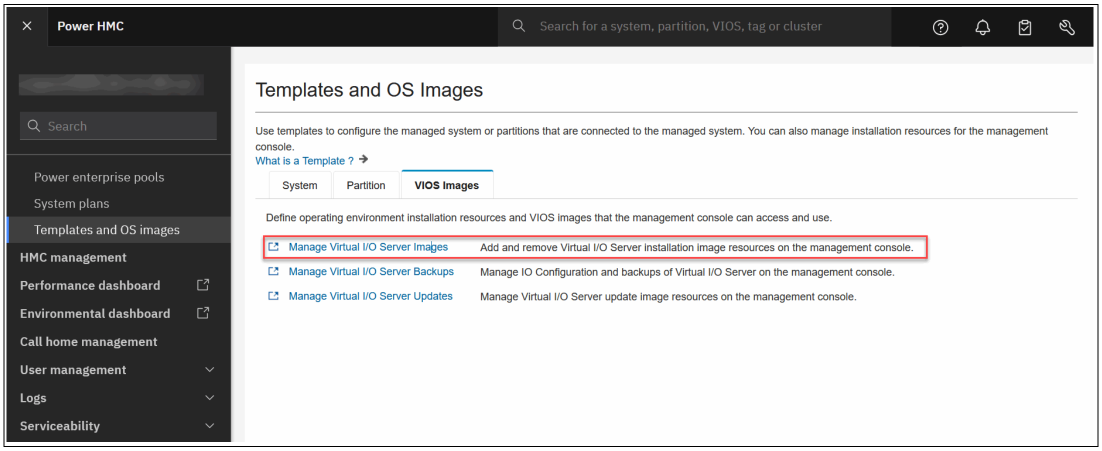
For maintenance of the Virtual I/O Server, VIOS might have to be powered off, which might impact the client partition to which the VIOS is providing a storage or a network. Therefore, before a VIOS is powered off, it is recommended that an user first validate redundancy of the dual-VIOS configuration, ensuring that the client partitions are properly setup with multi-path/redundancy for network & storage devices.

The HMC enhanced interface provides a wizard which helps with the process.

To learn more about the functionality see Prepare VIOS for Maintenance and other PowerVM management enhancements.

5.2.6 Installation of Virtual I/O Server images from the HMC

Due to PowerVM simplification it is possible to store VIOS images in the HMC repository. The stored images can be used for the VIOS installation, directly from the HMC. The image can be imported from a remote location, a DVD media or an USB.



Templates and OS Images

Use templates to configure the managed system or partitions that are connected to the managed system. You can also manage installation resources for the management console.

What is a Template ?

Define operating environment installation resources and VIOS images that the management console can access and use.

Manage Virtual I/O Server Images	Add and remove Virtual I/O Server installation image resources on the management console.
Manage Virtual I/O Server Backups	Manage IO Configuration and backups of Virtual I/O Server on the management console.
Manage Virtual I/O Server Updates	Manage Virtual I/O Server update image resources on the management console.

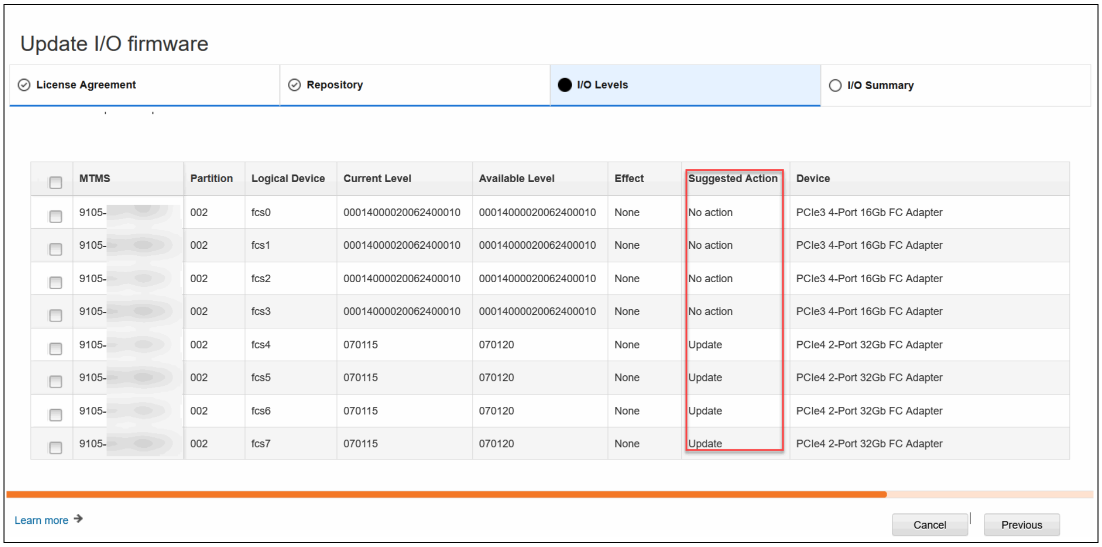
Figure 5-6 Virtual I/O Server image wizard

5.2.7 Microcode code update in I/O adapters

While Virtual I/O Server is an important component in the PowerVM environment, it is important to understand and plan for regular microcode updates to I/O adapters which belong to the VIOS configuration.

PowerVM simplification allows to perform microcode upgrade directly from the HMC GUI. The wizard automatically recognizes the latest available level on IBM website, and provide list of adapters which require update.

Figure 5-7 shows the simplified microcode upgrade function.



□	MTMS	Partition	Logical Device	Current Level	Available Level	Effect	Suggested Action	Device
□	9105-[redacted]	002	fcs0	00014000020062400010	00014000020062400010	None	No action	PCIe3 4-Port 16Gb FC Adapter
□	9105-[redacted]	002	fcs1	00014000020062400010	00014000020062400010	None	No action	PCIe3 4-Port 16Gb FC Adapter
□	9105-[redacted]	002	fcs2	00014000020062400010	00014000020062400010	None	No action	PCIe3 4-Port 16Gb FC Adapter
□	9105-[redacted]	002	fcs3	00014000020062400010	00014000020062400010	None	No action	PCIe3 4-Port 16Gb FC Adapter
□	9105-[redacted]	002	fcs4	070115	070120	None	Update	PCIe4 2-Port 32Gb FC Adapter
□	9105-[redacted]	002	fcs5	070115	070120	None	Update	PCIe4 2-Port 32Gb FC Adapter
□	9105-[redacted]	002	fcs6	070115	070120	None	Update	PCIe4 2-Port 32Gb FC Adapter
□	9105-[redacted]	002	fcs7	070115	070120	None	Update	PCIe4 2-Port 32Gb FC Adapter

Figure 5-7 Microcode update wizard

5.2.8 NFS Mounted ISOs in the Virtual Media Library

VIOS 4.1.1.00 Fix Pack Release also comes with the capability of mounting ISO images from an NFS server that appear as installable media from within the Virtual Media Library.

VIOS 4.1.1 adds support for NFS Mounted ISOs in Virtual Media Library which allows you to load ISO images from a centralized NFS server, eliminating need for repeated copying of ISO images across multiple VIOS. This saves storage space and time, maintains consistent images, and supports both NFS V3 and V4, allowing multiple images to be linked into the repository.

The **mkvopt** command is enhanced to support the new option –nfslink which creates a symbolic link to the specified NFS ISO file, in the repository. This is shown 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
```

You can find more information on this new capability in this IBM PowerVM Community blog.

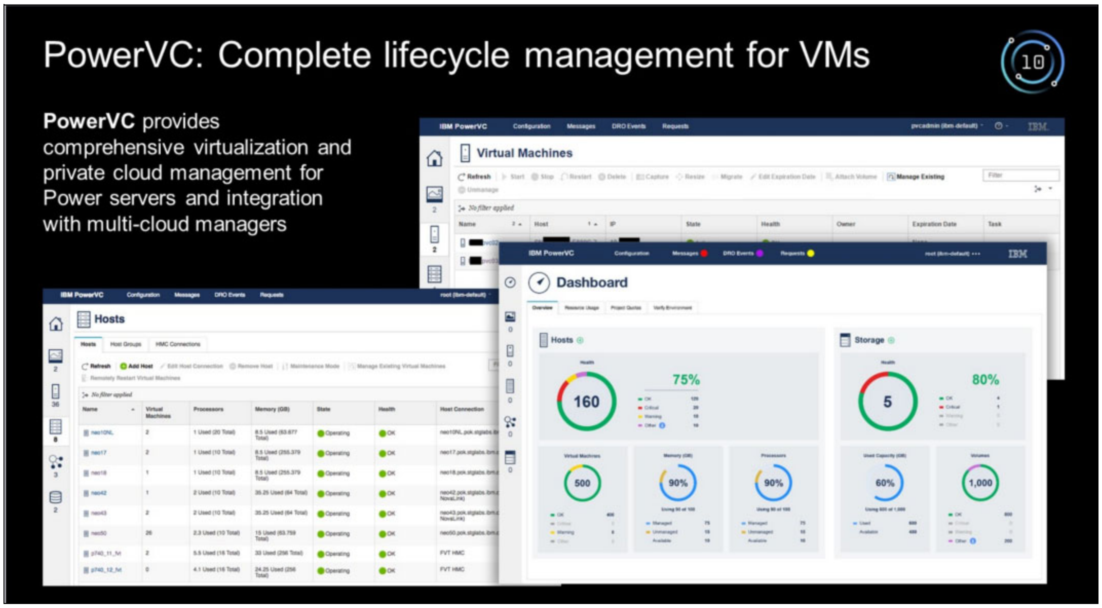
5.3 Power Virtualization Center overview

IBM Power Virtualization Center (PowerVC) is a robust cloud and virtualization management solution tailored for IBM Power Systems. It streamlines the management of virtual machines (VMs) across IBM Power Systems, simplifying tasks such as creating, deploying, resizing, and migrating VMs. PowerVC enables organizations to build and manage private cloud environments on their existing Power Systems 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.

With PowerVC, organizations can reduce total cost of ownership by simplifying cloud deployments and optimizing the movement of workloads and policies. This helps maximize resource utilization while delivering an intuitive user experience.

PowerVC captures and manages essential infrastructure details, such as VM definitions, storage, networking, and server configurations. This allows IT teams to maintain a library of VM images, facilitating rapid deployment by launching pre-configured images instead of 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 efficiency by quickly adjusting to workload demands. This flexibility helps reduce administrative costs, while making IT departments more agile in responding to business needs and market trends. Figure 5-8 shows how PowerVC can improve your client experience.



PowerVC: Complete lifecycle management for VMs

PowerVC provides comprehensive virtualization and private cloud management for Power servers and integration with multi-cloud managers

Figure 5-8 PowerVC management capabilities

Key benefits of PowerVC include:

- Seamless installation and configuration of the entire hardware stack (host, storage, network) and software components.
- Rapid setup, with installation possible within hours, regardless of administrator skill level.
- A small footprint with a streamlined client experience across the entire lifecycle, from deployment to ongoing operations and support.
- A reliable, cost-effective, and extensible platform for virtualization and cloud capabilities on Power Systems.
- Industry-standard APIs that support easy integration with higher-level cloud services.
- Efficient management of workloads with policy-based optimization, dynamically adjusting resources or moving workloads to under utilized systems.

For more information about PowerVC see Introduction to PowerVC for Private Cloud in IBM online documentation.

PowerVC functions

PowerVC provides the ability to perform the following functions with IBM PowerVM:

- Dynamically deploy VM OpenStack Images with Storage and Network resources.
- Dynamically resize the VMs CPU and memory using custom compute templates (OpenStack flavors).
- Dynamically assign disk volumes from predefined storage connectivity groups.
- Dynamically assign new network interfaces from pre-defined networks.
- Import existing VMs visible to the HMC to be managed by IBM PowerVC.
- Monitor and Manage resource usage with project namespaces and quotas.
- Create a snapshot or clone an existing VM image for subsequent deployment or backup.
- Migrate VMs between managed systems using Live Partition Mobility (LPM).
- Remote 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 expense and power consumption.

PowerVC utilizes both the IBM PowerVM Dynamic Partitioning abilities of 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.

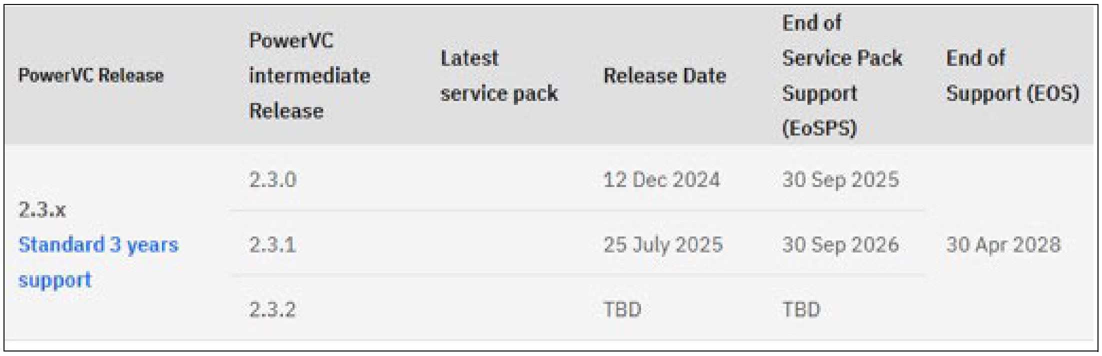
PowerVC offerings

IBM Power Virtualization Centre (PowerVC) is built on OpenStack. It provides simplified virtualization management and cloud deployments for IBM AIX, IBM i and Linux virtual machines (VMs) running on IBM Power. The offering is designed to build private cloud capabilities on Power servers and improve administrator productivity. It can further integrate with cloud environments through higher-level cloud orchestrators.

PowerVC is available in two editions:

- – 5765-VCC IBM Cloud PowerVC Manager
- – 5765-VC2 PowerVC for Private Cloud

The main difference between versions is that PowerVC for Private Cloud enables a self-provisioning function for project users. The most recent version of PowerVC is 2.3.x. Figure 5-9 shows the recent release history for PowerVC as of the publication of this document. For the most current information see `https://www.ibm.com/support/pages/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	30 Apr 2028
	2.3.1		25 July 2025	30 Sep 2026	
	2.3.2		TBD	TBD	

Figure 5-9 PowerVC 2.3 service history

Note: For more details on what is new in IBM PowerVC for Private Cloud see https://www.ibm.com/docs/en/powervc-cloud/2.3.0?topic=introduction-whats-new-in-powervc-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 or virtual machine. This approach simplifies installation and management, making it suitable for smaller environments or proof-of-concept deployments. While offering ease of use and reduced complexity, the single-node architecture inherently limits scalability and high availability, as a single point of failure can disrupt the entire PowerVC environment. Therefore, while convenient for initial setups, it becomes less ideal as the data center grows and demands increased resilience and capacity.

High availability configuration

Early versions of IBM PowerVC employed an All-in-One (AIO) deployment, consolidating all necessary packages and services—both stateless (OpenStack-based) and stateful—onto a single system. While this simplified deployment and management, it restricted PowerVC's ability to scale and achieve high availability, crucial for larger data centers with numerous IBM Power servers.

Recognizing this limitation, PowerVC evolved to support a multi-node architecture, addressing the need for both scalability and resilience. This shift aims to eliminate single points of failure by providing a highly available management node, ensuring that the failure of one node does not disrupt virtualization and cloud management. Furthermore, distributing the workload across multiple nodes enables PowerVC to handle increased scale and support larger deployments. PowerVC 2.0.2 introduced a three-node architecture, laying the foundation for a highly available and scalable solution designed to meet the demands of enterprise-level data centers. The HA configuration is shown in Figure 5-10.

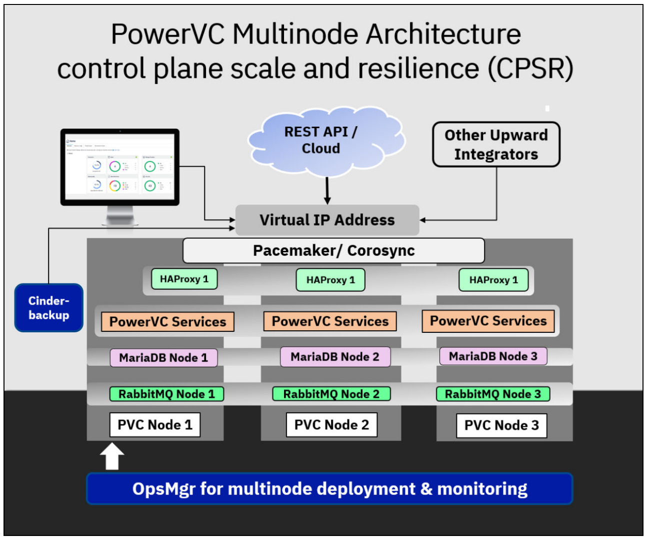


Figure 5-10 PowerVC multinode implementation

Compute nodes for additional scalability

Compute plane nodes (CPNs) in PowerVC are designed to enhance scalability and reliability by distributing the compute services load, particularly when managing hosts through the Hardware Management Console (HMC).

Traditionally, each HMC-managed host triggered the startup of a dedicated compute service on the PowerVC management server, limiting the number of manageable hosts and necessitating multiple PowerVC instances for larger environments. CPNs address this limitation by allowing users to register standard RHEL or SLES virtual machines within the same subnet as the PowerVC management node. Once registered, these CPNs take on the responsibility of starting compute services, effectively offloading the management server and enabling the management of a greater number of hosts.

This approach eliminates the need for multiple PowerVC instances and provides two distinct methods for scaling compute hosts, significantly 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 host in the PowerVC instance.

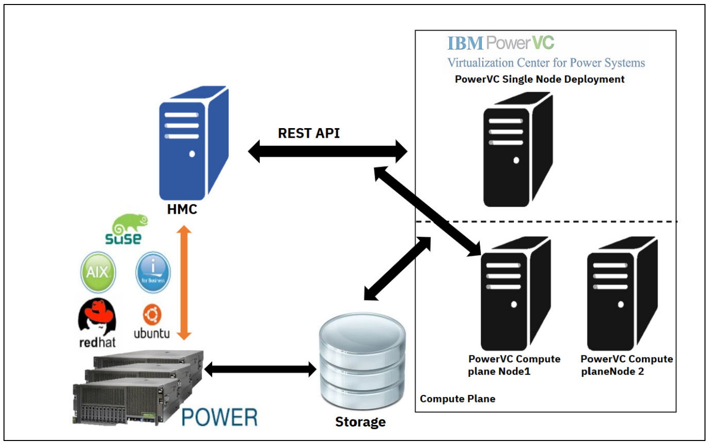

IBM PowerVC
Virtualization Center for Power Systems
PowerVC Single Node Deployment
REST API
HMC
suse
AIX
redhat
ubuntu
POWER
Storage
PowerVC Compute plane Node1
PowerVC Compute planeNode 2
Compute Plane

Figure 5-11   Single node PowerVC with CPNs

Figure 5-11 shows the addition of CPNs to a PowerVC HA multinode implementation to provide both scalability and high availability.

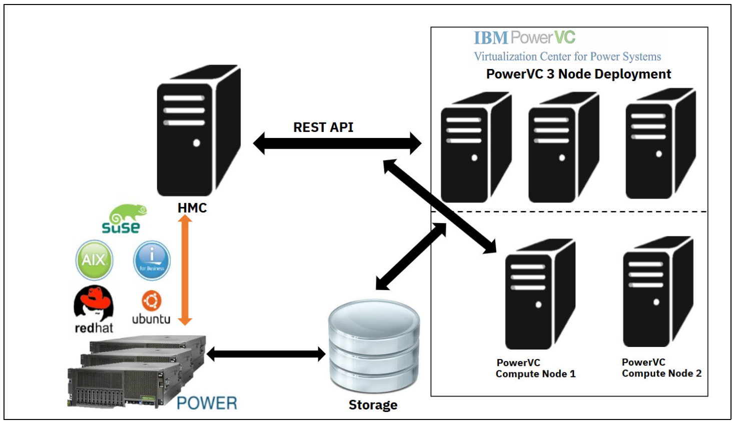


Figure 5-12   HA implementation of PowerVC with CPNs

5.3.2 PowerVC connectivity to managed systems

Using PowerVC, there are two options to manage Power servers. PowerVC can either use an HMC to manages a server or a NovaLink LPAR. If Novalink is used, a special partition on each Power server is needed to provide the management functions as is done in the HMC. A combined solution is also possible, where both HMC and NovaLink exist together.

HMC management

The IBM Hardware Management Console (HMC) is a hardware appliance, or it can be a virtual appliance, that serves as a vital tool for managing IBM Power Systems servers. It provides system administrators with the ability to configure and control one or more managed systems, including the creation and management of logical partitions (LPARs), and the activation of Capacity Upgrade on Demand. Beyond partitioning, the HMC facilitates essential service functions, enabling the detection, consolidation, and transmission of system information to IBM service and support. Essentially, it acts as a central control point for Power Systems hardware, ensuring stability, efficient resource allocation, and streamlined service operations.

PowerVC uses rest API calls to connect to an HMC to create or modify LPARs on an HMC managed system. If you typically use redundant HMCs to manage your hosts, you can continue the practice with PowerVC. Each host can be managed by multiple HMCs. The HMC through which the host is registered first will be the primary HMC for the host. If the host is also being managed by a second HMC and that HMC is later added to PowerVC, then that HMC is set as secondary HMC for the host in PowerVC. If the primary HMC fails, PowerVC automatically fails over the host connection to the secondary HMC.

PowerVC can manage up to 45 HMC-managed hosts when using the new compute node function. Each host can have a maximum of 1000 virtual machines on it with a maximum of 5000 virtual machines on all of the hosts combined. Each HMC can manage a maximum of 2000 virtual machines.

NovaLink

PowerVM NovaLink is a software interface that is used for virtualization management. Now PowerVM NovaLink software version 2.3.0 installs on a Red Hat Enterprise Linux (RHEL) partition 8.10 or 9.2, or later. NovaLink enables highly scalable modern cloud management and deployment of critical enterprise workloads.

The PowerVM NovaLink installer provides the rapid provisioning of large numbers of virtual machines. An installer boot from the System Management Service (SMS) interface of the Power system creates a RHEL PowerVM NovaLink partition and Virtual I/O Server (VIOS) partitions, installing operating systems and the PowerVM NovaLink software. The PowerVM NovaLink installer reduces the installation time and facilitates repeatable deployments, which is particularly desirable in large scale out implementations.

NovaLink runs in a partition on each Power server being managed by NovaLink. This is shown in Figure 5-13 on page 117. If you install the PowerVM NovaLink environment on a new managed system, the PowerVM NovaLink installer creates the PowerVM NovaLink partition automatically. If you are installing NovaLink on an HMC managed system, create a Linux partition and then use the NovaLink installer to install the partition.

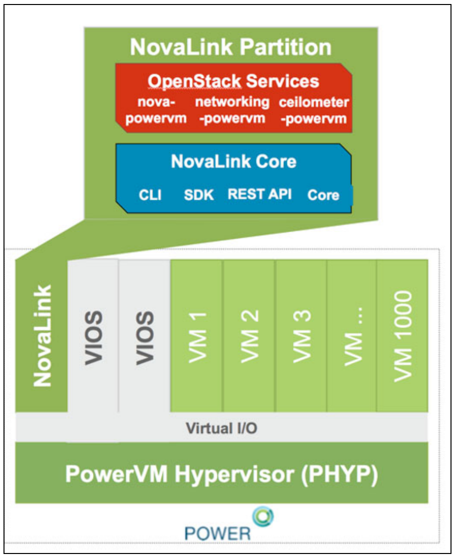


Figure 5-13 NovaLink architecture

PowerVM Novalink, reduces the complexity and increases the security of your server management infrastructure. PowerVM NovaLink provides a server management network interface on the server. The server management network between PowerVM NovaLink and its virtual machines is secure by design and is configured with minimal user intervention.

Utilizing NovaLink provides the highest scalability in terms of the number of systems that can be managed by PowerVC. A maximum of 60 NovaLink-managed hosts is supported per PowerVC instance. A maximum of 10000 virtual machines and 20000 volumes can be on all of the NovaLink-managed hosts combined. A maximum of 1000 virtual machines (NovaLink, Virtual I/O Servers, or client workloads) per PowerVM host are supported. This limit is determined by the PowerVM platform firmware versions available when PowerVC version 2.3.0 was released.

Novalink works with PowerVC or other OpenStack solutions to manage your Power servers in the hybrid cloud. For more information refer to 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 on page 118. When a managed system is co-managed by the HMC and PowerVM NovaLink, you set either PowerVM NovaLink or the HMC to be in the 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 the controller mode, you can run partition change operations only by using PowerVM NovaLink. If you want to run 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 set to the controller mode to update the firmware. For more information see Managed systems co-management.

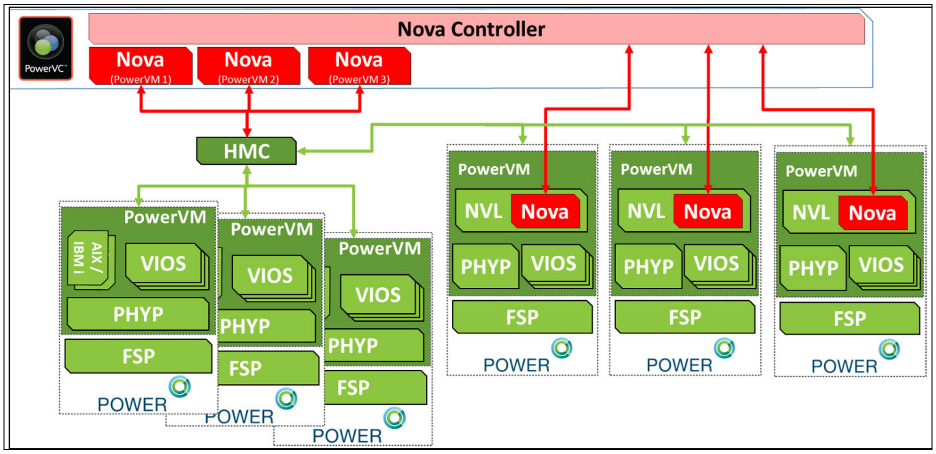


Figure 5-14 PowerVC with HMC and Novalink partitions

When your environment consists of both HMC-managed and NovaLink-managed hosts, a maximum of 3000 virtual machines can be on all of the hosts combined. A maximum of 50 hosts is supported of which 30 hosts can be HMC-managed.

PowerVC OpenStack

PowerVC for Private Cloud 2.3.0 is built on OpenStack Caracal and supports Python3.11. OpenStack Caracal 2024.1 adds enhancements for AI and HPC; improves agility, performance, and security; and incorporates support for industry-best hardware and software.

> **Note:** For more details on the OpenStack Caracal release see https://www.openstack.org/software/openstack-caracal/

PowerVC streamlines the management of IBM Power Systems by offering two primary connection methods: it can either connect directly to the Hardware Management Console (HMC), which then acts as a proxy for accessing the managed systems, or it can connect directly to NovaLink partitions on those systems. This flexibility allows PowerVC to rapidly discover and understand the PowerVM configuration of the managed systems, including the Virtual I/O Servers (VIOS) and their connections to SAN fabric and Ethernet networks. This immediate discovery minimizes the need for extensive manual configuration by administrators, simplifying and accelerating the setup and management of virtualized environments.

The OpenStack software has industry-standard interfaces that are released under the terms of the Apache License. IBM Power Virtualization Center interfaces are a subset of OpenStack northbound APIs. A number of 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 APIs provided by OpenStack and can be used with PowerVC without any modifications.
- Extended OpenStack APIs - These APIs are a subset of the APIs provided by OpenStack, but their functions are extended by PowerVC.
- PowerVC APIs - These APIs do not exist in OpenStack and are exclusive to PowerVC.

Supported OpenStack APIs can be accessed using Ansible OpenStack modules. Providing a mechanism to fully automate the administration and deployment of VM resources within the IBM Power Managed system infrastructure. Figure 5-15 shows the different automation options for configuring your IBM Power infrastructure.

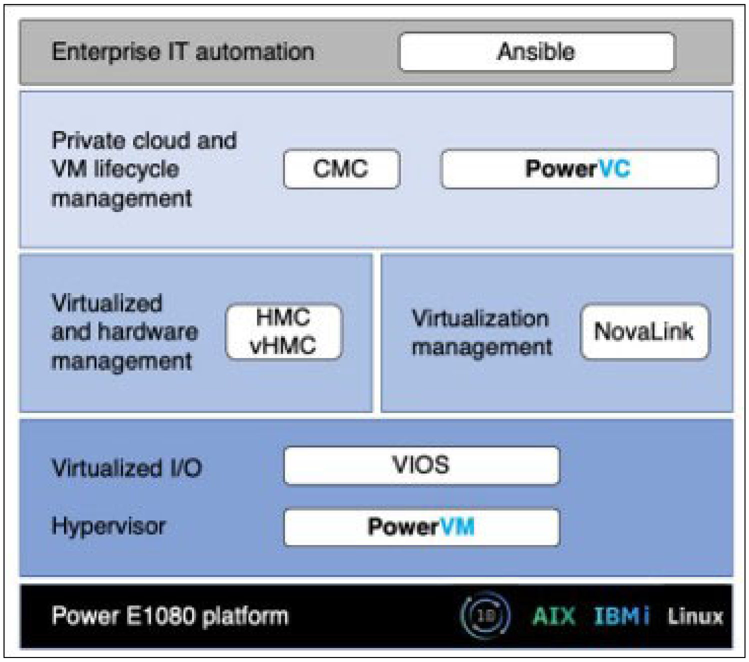


Figure 5-15 The Power Automation Stack

5.3.3 PowerVC automation

IBM PowerVC is an advanced virtualization and cloud management offering. Built on OpenStack, it provides simplified virtualization management and cloud deployments for IBM AIX, IBM i and Linux virtual machines (VMs) running on IBM Power Systems. The offering is designed to build private cloud capabilities on Power Systems servers and improve administrator productivity. It can further integrate with cloud environments through higher-level cloud orchestrators.

IBM Cloud Pak for AIOps is an AIOps platform that deploys advanced, explainable AI using the IT Operations (ITOps) toolchain data so that you can confidently assess, diagnose, and resolve incidents across mission-critical workloads.

IBM Cloud Pak for AIOps powers automation by using diverse data sets from an entire range of hybrid environments from cloud to on-premises, and bringing the information together across ITOps. With this Cloud Pak, you can tap into shared automation services to get insight into how your processes run. You can also visualize hotspots and bottlenecks, and pinpoint what to fix with event detection to prioritize which issues to address first.

Infrastructure automation is included as an optional feature with IBM Cloud Pak for AIOps and is installed within a separate OpenShift cluster from IBM Cloud Pak for AIOps. The Infrastructure Automation plugin supports IBM PowerVC v1.4.4 or higher.

In Infrastructure Automation, a manager is an external management environment that manages more than one type of resource. One example of a manager is the PowerVC provider, which manages infrastructure, cloud, network, and storage resources.

Infrastructure automation consists of the following components, which were available with IBM Cloud Pak for Multicloud Management.

- Infrastructure management, previously called IBM Red Hat CloudForms.

  Infrastructure management delivers the insight, control, and automation enterprises need to address the challenges of managing virtual environments, which are far more complex than physical ones. This technology enables enterprises with existing virtual infrastructures to improve visibility and control, and those just starting virtualization deployments to build and operate a well-managed virtual infrastructure.

- Managed services, previously called Terraform & Service Automation or IBM Cloud Automation Manager.

  Managed services provide you with the capability to automate provisioning of infrastructure and virtual machine applications across multiple cloud environments with optional workflow orchestration.

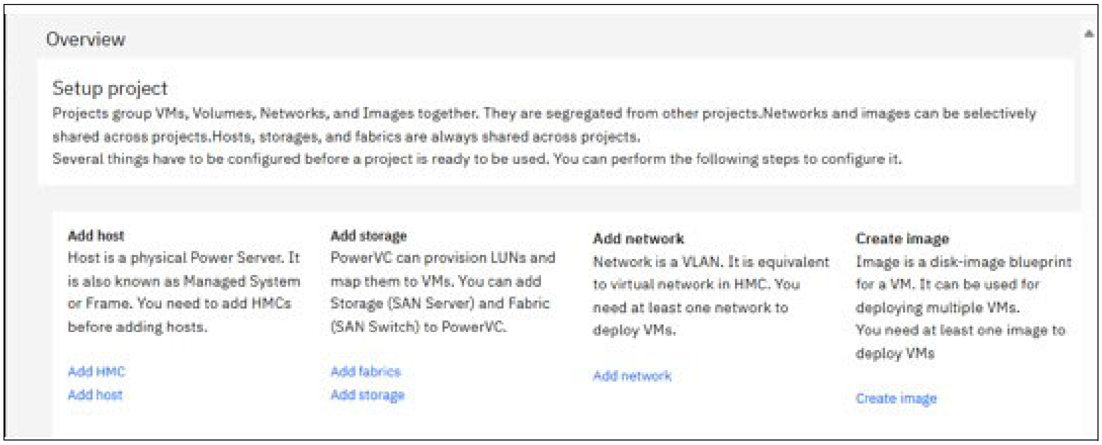
Infrastructure automation enables IT Operations and Site Reliability Engineer (SRE) teams to use infrastructure as code practices to drive IT velocity and shift to the left of operations. Providing Terraform and ManageIQ for an integrated, Infrastructure Automation capability for IT Operators.

When you have a license for IBM Cloud Pak for AIOps, you are entitled to install and use Infrastructure Automation.

> **Note:** Learn how to integrate PowerVC into IBM Cloud Paks for AIOps
> `https://www.ibm.com/docs/en/cloud-paks/cloud-pak-aiops/4.8.1?topic=providers-powervc`

5.3.4 PowerVC setup tasks

Before you can use PowerVC there are number of configuration tasks that need to be performed. These are shown in Figure 5-16.



Overview

Setup project

Projects group VMs, Volumes, Networks, and Images together. They are segregated from other projects.Networks and images can be selectively shared across projects.Hosts, storages, and fabrics are always shared across projects.
Several things have to be configured before a project is ready to be used. You can perform the following steps to configure it.

Add host
Host is a physical Power Server. It is also known as Managed System or Frame. You need to add HMCs before adding hosts.

Add storage
PowerVC can provision LUNs and map them to VMs. You can add Storage (SAN Server) and Fabric (SAN Switch) to PowerVC.

Add network
Network is a VLAN. It is equivalent to virtual network in HMC. You need at least one network to deploy VMs.

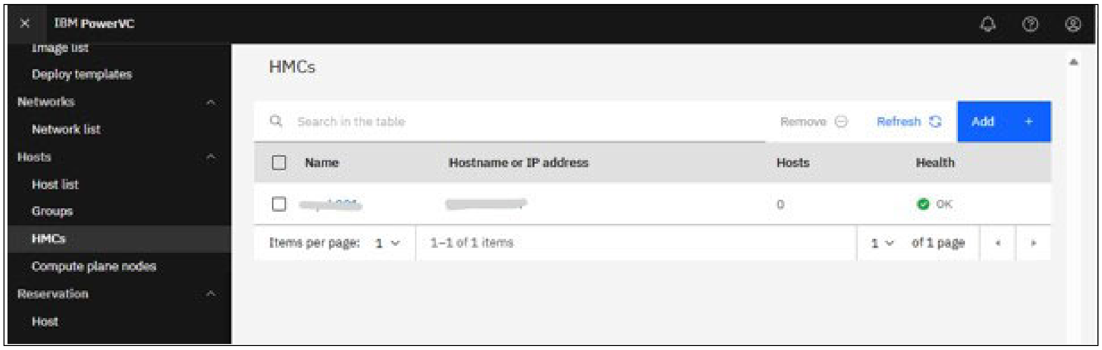
Create image
Image is a disk-image blueprint for a VM. It can be used for deploying multiple VMs.
You need at least one image to deploy VMs

Figure 5-16 Initial screen when logging into a new installation

All configuration tasks are available in the PowerVC GUI.

Add HMC

Add the HMC that is attached to the IBM Power Managed Systems as shown in Figure 5-17.



☐	Name	Hostname or IP address	Hosts	Health
☐	[redacted]	[redacted]	0	✓ OK

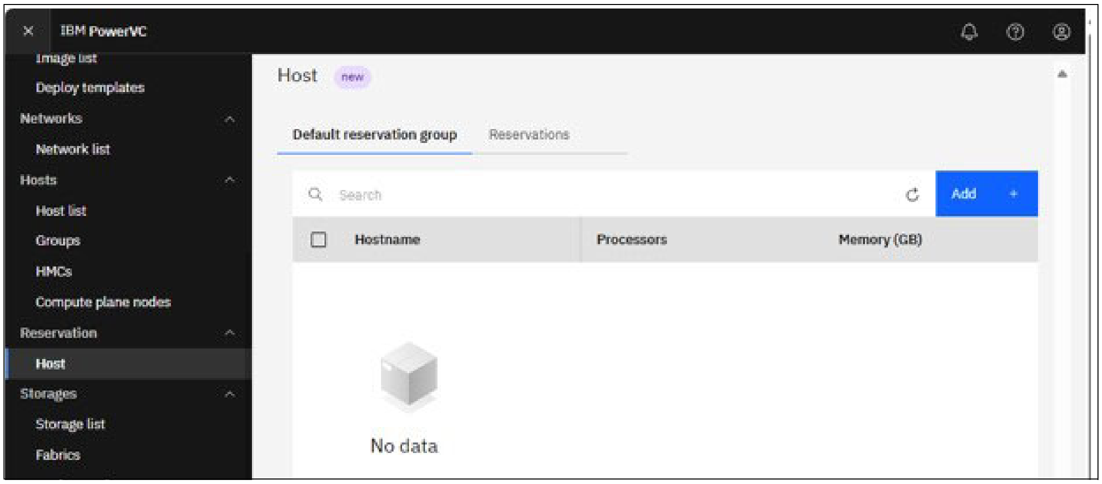
Items per page: 1 1–1 of 1 items 1 of 1 page

Figure 5-17 Adding HMC to PowerVC

The HMC connection and discovery process already has all the information of the connected IBM Power systems. PowerVC just needs to get authorization for the HMC connection.

Add host

Once PowerVC knows about an HMC, we can add a host definition to the PowerVC database as shown in Figure 5-18. This is a discovery process where PowerVC imports the managed system information into the PowerVC database.



☐	Hostname	Processors	Memory (GB)

No data

Figure 5-18 Add host

Add fabric

A discovery process through the connected SAN storage Fabric where the VIOS Host Bus Adapters (HBAs) are zoned.

This task would be accomplished with the storage administrator, who may provide a separate PowerVC userid with admin privileges to the SAN storage, accessing a defined storage pool for PowerVC deployments.

PowerVC will also has the ability to discover existing storage, and existing VMs using the storage. Figure 5-19 shows adding a fabric through the GUI.

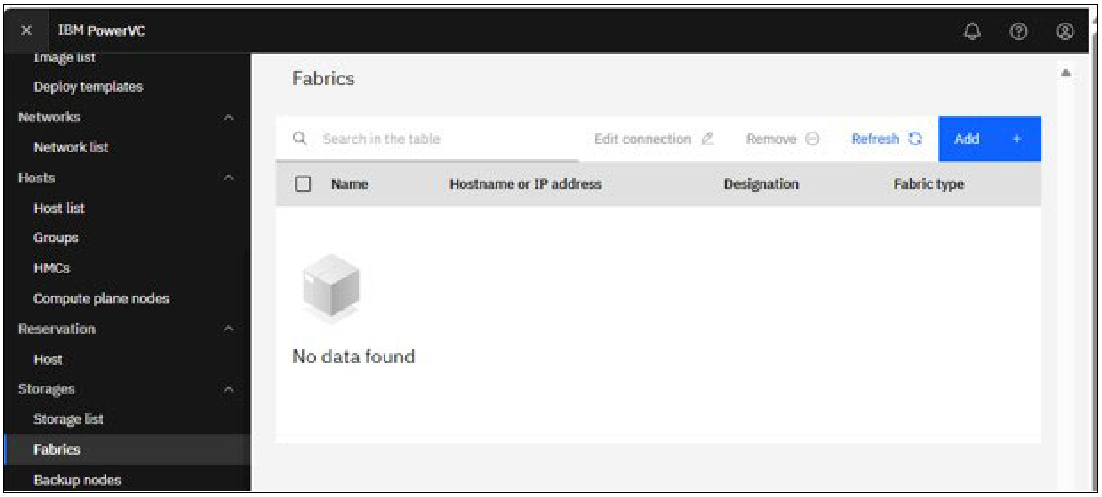
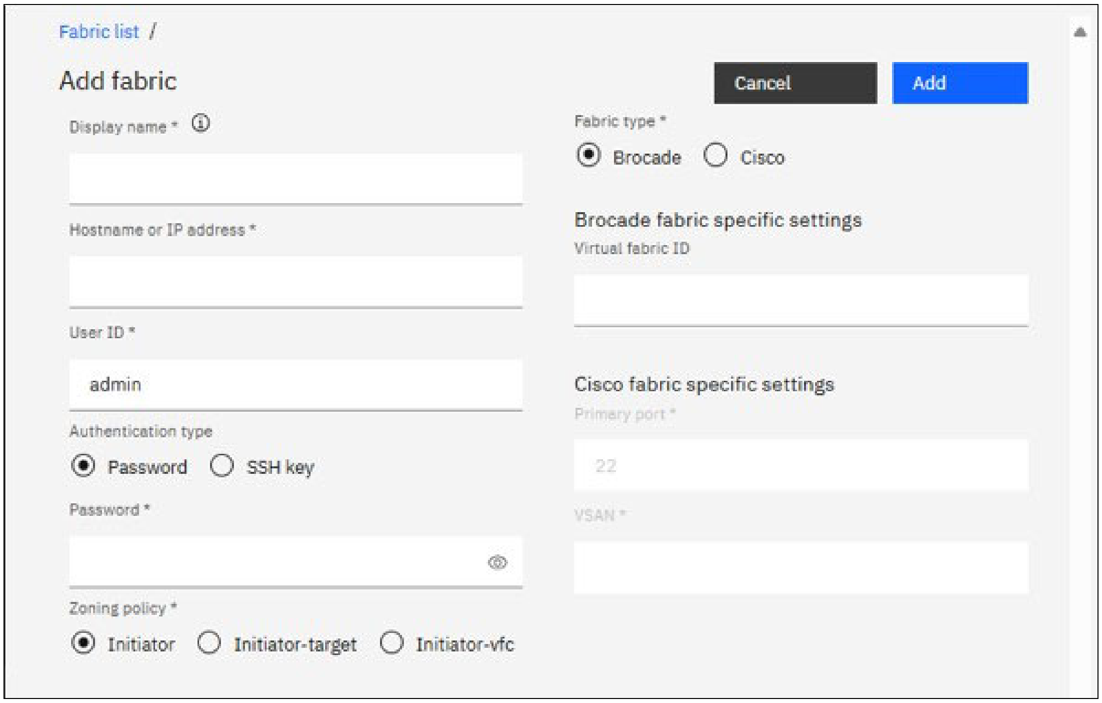


Figure 5-19 Adding a fabric

Additional details are required to determine the type of Fabric configuration that will be connected to as shown in Figure 5-20.



Fabric list /

Add fabric

Cancel Add

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 *

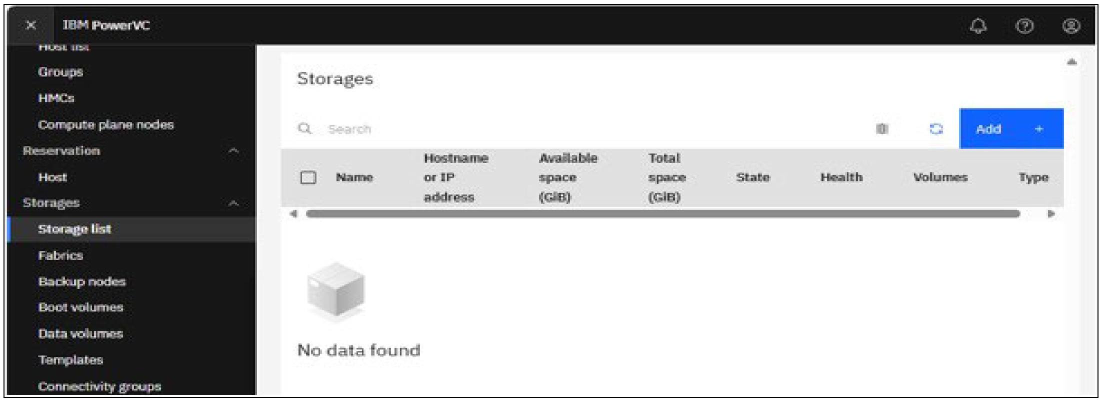
22

VSAN *

Figure 5-20 Add fabric details

Add storage

Once access to the fabric is available, another discovery process is used to add a storage system in the SAN fabric as shown in Figure 5-21.



Name	Hostname or IP address	Available space (GiB)	Total space (GiB)	State	Health	Volumes	Type

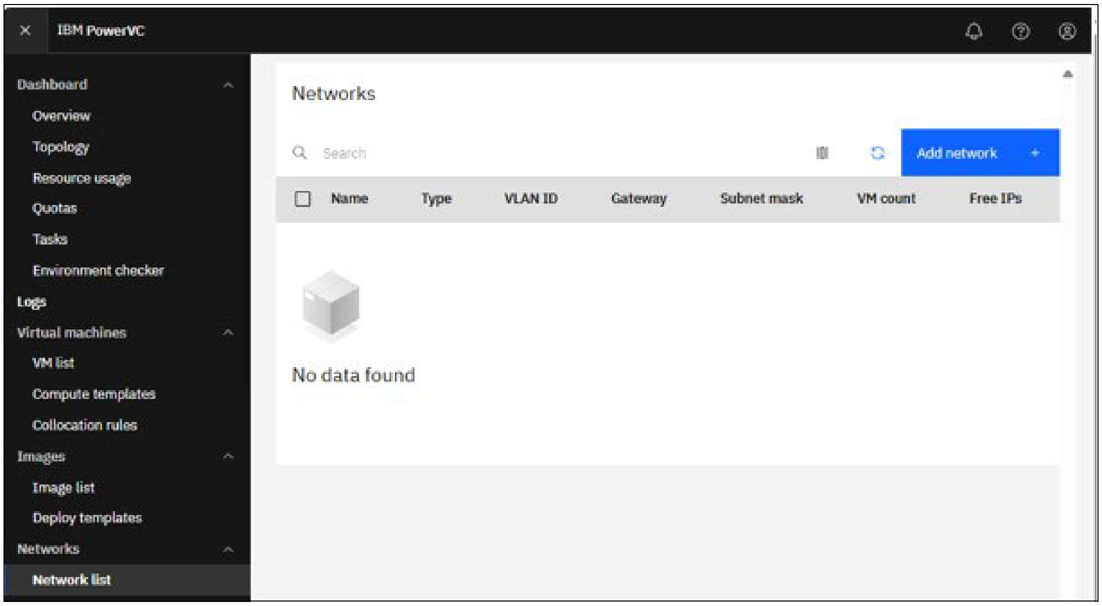
No data found

Figure 5-21 Add storage controller

There are other more detailed storage functions possible, for example, setting up LUN replication between storage systems. Implementing these additional functions are made easier if standard zoning practices are set up and may require discussions with the local storage administrator. Ensure that the SAN fabric zoning is compatible with Live Partition Mobility zoning requirements.

Add Network

There is also a task for discovering any Shared Ethernet Adapter virtual networks available on the virtual i/o server (VIOS) as shown in Figure 5-22.

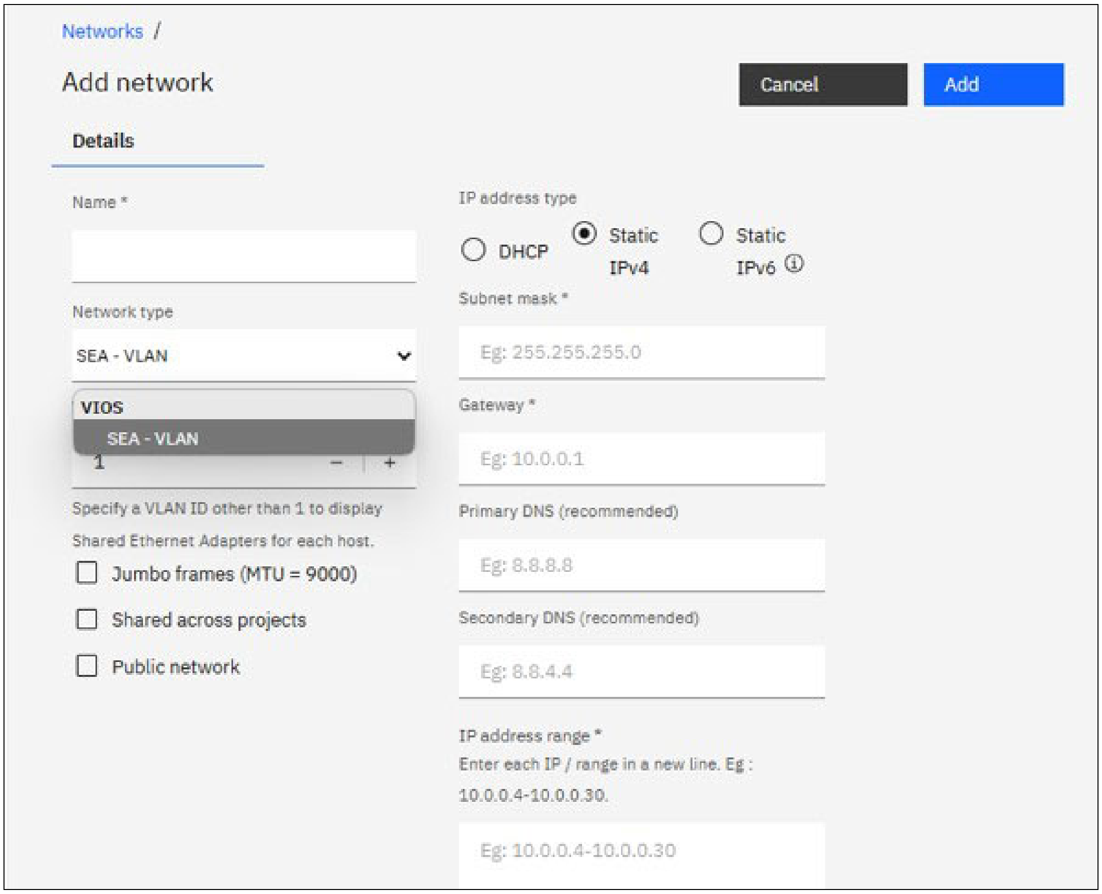


Name	Type	VLAN ID	Gateway	Subnet mask	VM count	Free IPs

No data found

Figure 5-22 Adding network connectivity

When PowerVC discovers a network connection which has been implemented based on best practices utilizing a dual VIOS redundant configuration, PowerVC will check the virtual network setup and indicate that redundancy is available. Figure 5-23 on page 124 displays a configuration where only an SEA backed network is available to be configured. This view represents the add network screen before any servers have been added to PowerVC.



Add network

Name *

Network type: SEA - VLAN

1

Specify a VLAN ID other than 1 to display Shared Ethernet Adapters for each host.

□ Jumbo frames (MTU = 9000)

□ Shared across projects

□ Public network

IP address type: ○ DHCP ● Static IPv4 ○ Static IPv6

Subnet mask * Eg: 255.255.255.0

Gateway * Eg: 10.0.0.1

Primary DNS (recommended) Eg: 8.8.8.8

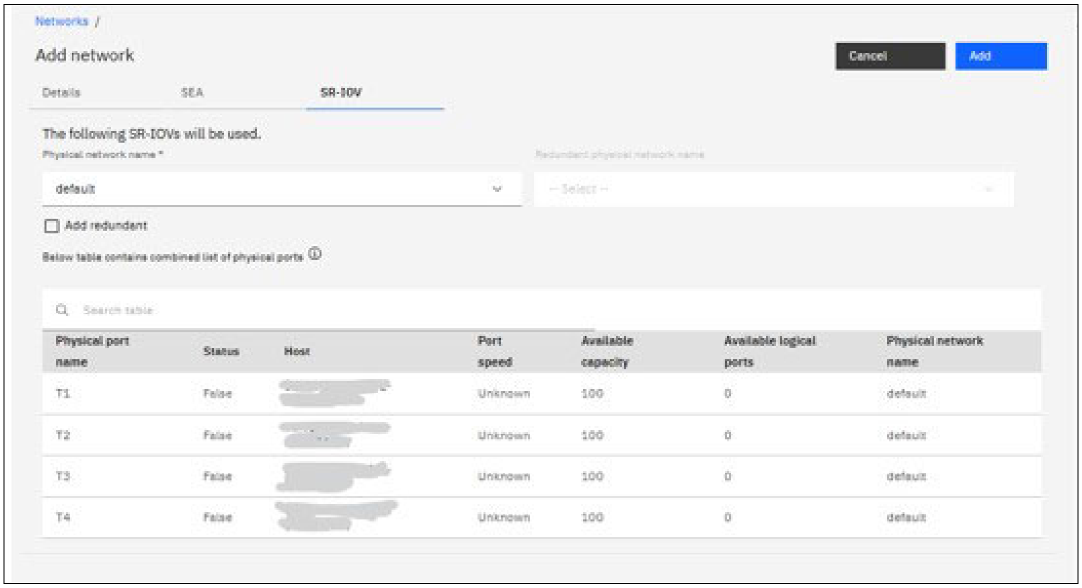
Secondary DNS (recommended) Eg: 8.8.4.4

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 Configure network connection

Once a managed system has been discovered, it’s capabilities will be known as indicated in Figure 5-24.



Add network

The following SR-IOVs will be used.

Physical network name *: default

Redundant physical network name: -- Select --

□ Add redundant

Below table contains combined list of physical ports

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

Preparing a Single Root Input/Output Virtualization (SRI/OV) adapter for sharing can be performed using the HMC GUI. For more details on the setup see https://www.ibm.com/docs/en/power10?topic=msia-modifying-sr-iov-adapters.

SR-IOV supports pass-through of Ethernet data from guest virtual machines directly to hardware. This improves performance by allowing data to pass directly from guest virtual machines to physical adapters with minimal processing involved allowing a guest virtual machine to achieve near wire-speed Ethernet performance. SR-IOV also supports some additional configuration options, such as quality of service (QoS) for enforcing bandwidth allocations to guest virtual machines.

PowerVC can connect a virtual machine that uses an SR-IOV adapter by using options 2, 3 or 4, depending on the specified redundancy levels. This is shown in Figure 5-25.

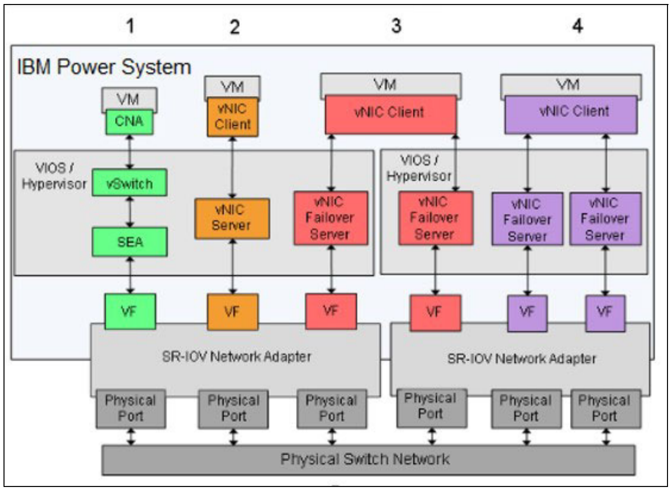


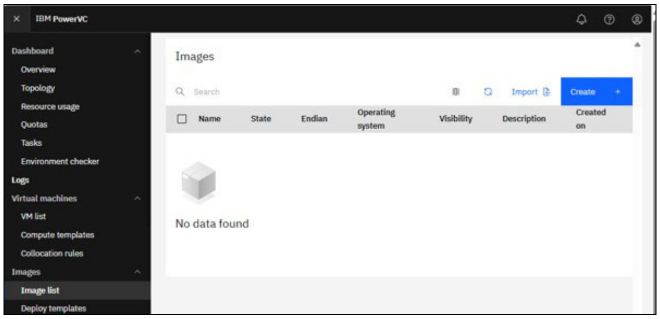
Figure 5-25 SR-IOV connectivity options

Note: For more information on using SR-IOV adapters with PowerVC see:

https://www.ibm.com/docs/en/powervc-cloud/2.3.0?topic=networks-sr-iov-backed
https://www.ibm.com/docs/en/power10?topic=msia-modifying-sr-iov-adapters

Add an Image

PowerVC enables you to capture and import images that you can deploy as virtual machines. An image consists of metadata and one or more binary images, one of which must be a bootable disk. To create a virtual machine in PowerVC, you must deploy an image. This is shown in Figure 5-26



□	Name	State	Endian	Operating system	Visibility	Description	Created on

No data found

Figure 5-26 Managing images

Before a virtual machine can be captured, it must meet specific requirements. If you do not prepare the virtual machine before you capture it, you can experience problems when you deploy the resulting image. For example, you might not be able to ping the virtual machine that is created when the image is deployed.

When capturing a virtual machine, all volumes that belong to its boot set are included in the image generated by the capture. If the virtual machine is brought into PowerVC management, then the boot set consists of all volumes that are marked as the boot set when managing the virtual machine. If the virtual machine is deployed from an image that is created within PowerVC, then the boot set consists of all volumes that the user chooses as the boot set when creating the image. Unlike the volumes that belong to the virtual machine's boot set, the user can choose which data volumes to include in the image generated by the capture.

You can use cloud-init to enable the virtual machines for capture. Cloud-init is a technology that takes user input and configures the operating system and software on deployed virtual machines. Cloud-init is widely used in OpenStack.

Note: For more information about working with images for PowerVC see:

https://www.ibm.com/docs/en/powervc-cloud/2.3.0?topic=administrator-working-images

Deploying a VM

Once PowerVC is set up and the required resources are defined, you can deploy a virtual machine. Using PowerVC you can deploy a VM and have it available for login within about 5 minutes, depending on network and storage connections. Figure 5-27 shows the list of VM deployment.

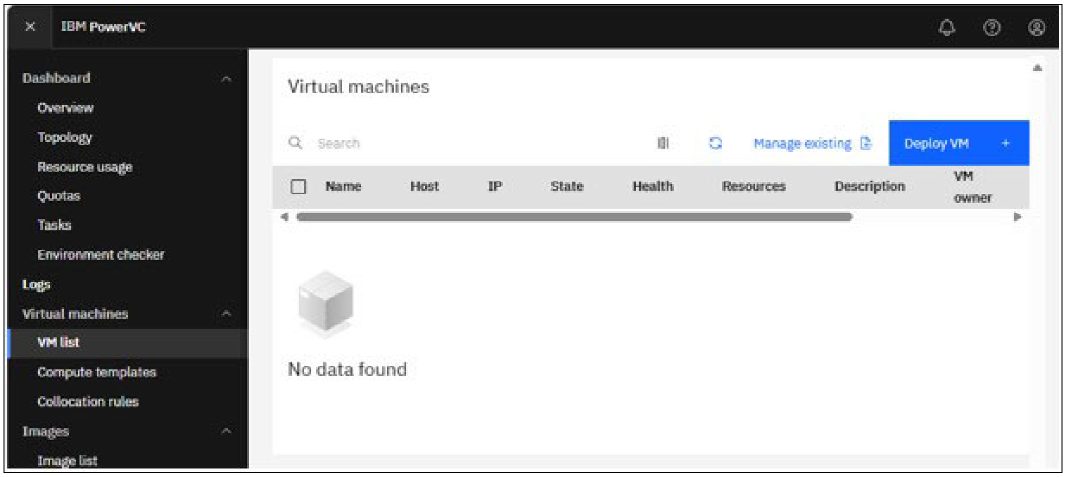


Figure 5-27 VM list

5.3.5 VM deployment and automation scenarios

Consider an infrastructure without PowerVC. Even if you used Ansible roles to automate the HMC setup including defining FC adapter assignments, you would still be left with a number of infrastructure tasks.

Figure 5-28 shows the remaining tasks, indicated in orange.

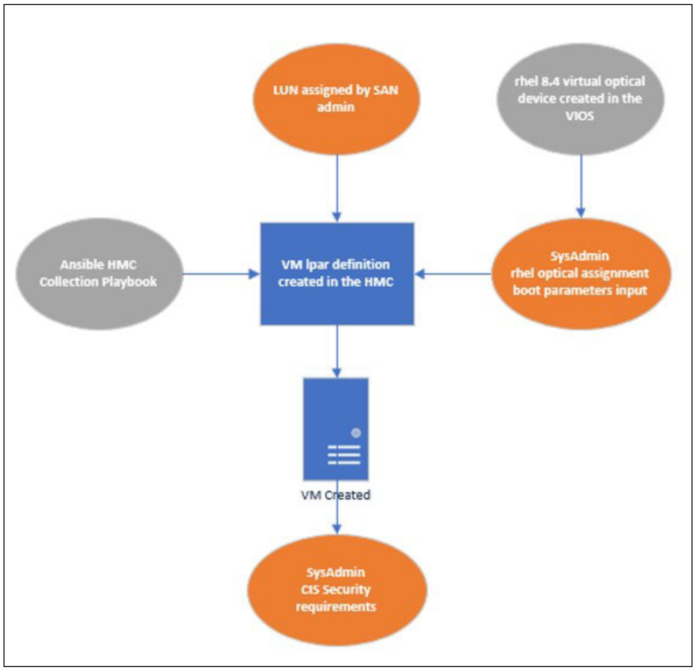

LUN assigned by SAN admin
rhel 8.4 virtual optical device created in the VIOS
Ansible HMC Collection Playbook
VM lpar definition created in the HMC
SysAdmin rhel optical assignment boot parameters input
VM Created
SysAdmin CIS Security requirements

Figure 5-28 deployment requires contact with SAN admin and security teams

We could try to automate additional setup using a kickstart file and tftpboot with a DHCP server. But we are still left with discussions with SAN admin, the security team and the network team for IP provisioning in DHCP as shown in Figure 5-29.

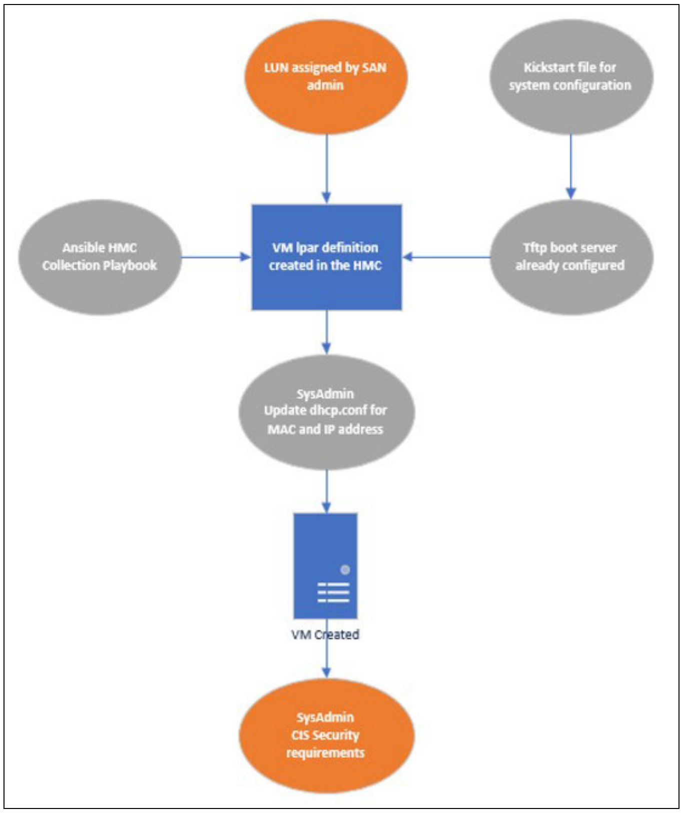

LUN assigned by SAN admin
Kickstart file for system configuration
Ansible HMC Collection Playbook
VM lpar definition created in the HMC
Tftp boot server already configured
SysAdmin Update dhcp.conf for MAC and IP address
VM Created
SysAdmin CIS Security requirements

Figure 5-29 Deployment with tftpboot, still requires the SAN admin and Security teams.

After implementing PowerVC to deploy a VM. We are ready to add Ansible to provision the remaining segments of a fully automated solution of VM deployment as shown in Figure 5-30.

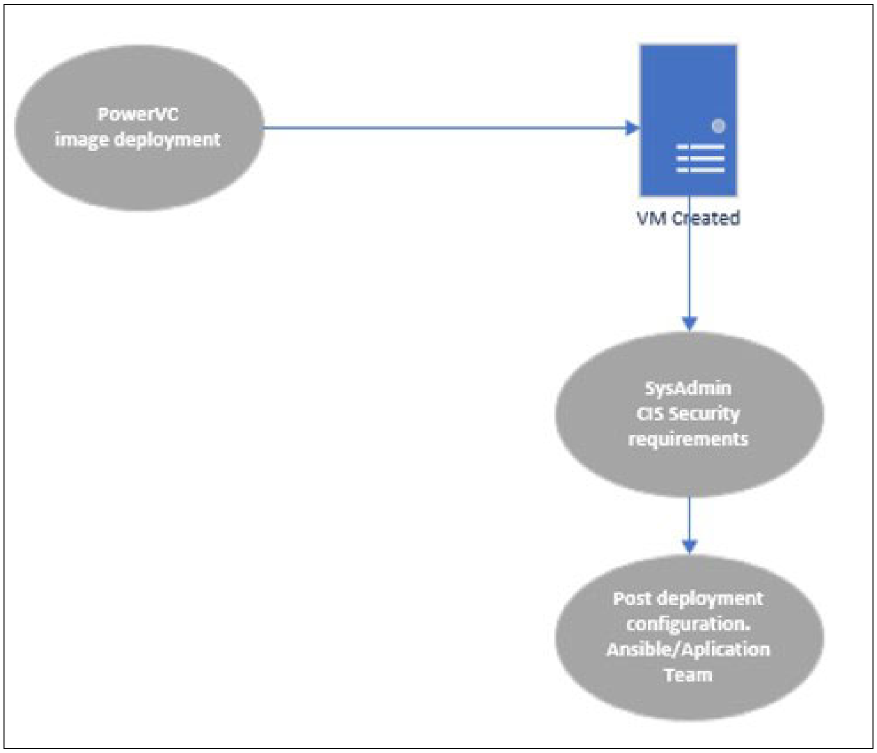
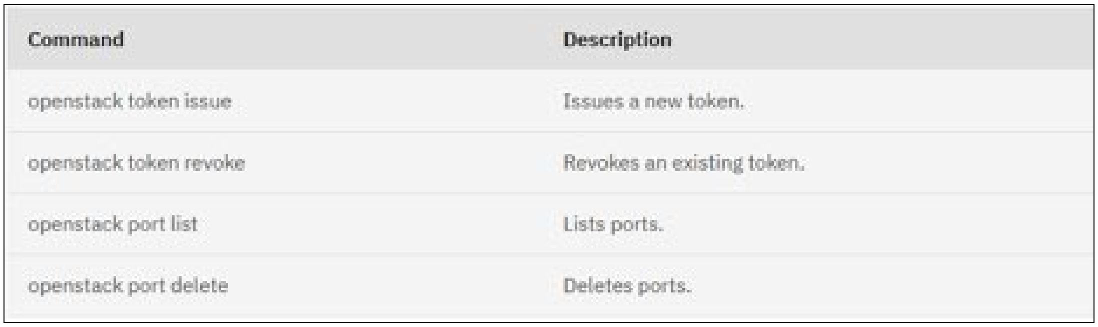


Figure 5-30 Deployment with PowerVC and Ansible

5.4 Ansible OpenStack Modules

PowerVC does not support all available OpenStack commands, only a sub-set of OpenStack commands are supported. Figure 5-31 shows some of the OpenStack commands.

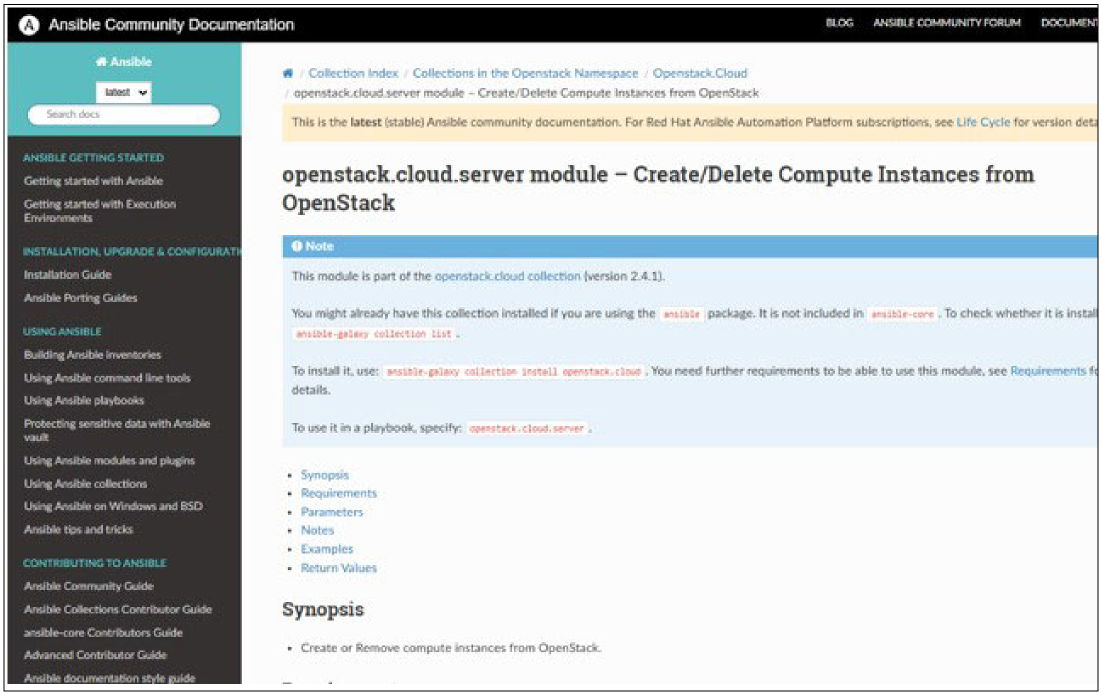


Command	Description
openstack token issue	Issues a new token.
openstack token revoke	Revokes an existing token.
openstack port list	Lists ports.
openstack port delete	Deletes ports.

Figure 5-31 Sample OpenStack commands

However, more OpenStack APIs are available using Ansible OpenStack Modules. The openstack.cloud collection provides the APIs needed to interact with IBM PowerVC.

Figure 5-32 shows the Ansible OpenStack cloud server module website.


Ansible Community Documentation
BLOG
ANSIBLE COMMUNITY FORUM
DOCUMENT
Ansible
latest
Search docs
ANSIBLE GETTING STARTED
Getting started with Ansible
Getting started with Execution Environments
INSTALLATION, UPGRADE & CONFIGURATI
Installation Guide
Ansible Porting Guides
USING ANSIBLE
Building Ansible inventories
Using Ansible command line tools
Using Ansible playbooks
Protecting sensitive data with Ansible vault
Using Ansible modules and plugins
Using Ansible collections
Using Ansible on Windows and BSD
Ansible tips and tricks
CONTRIBUTING TO ANSIBLE
Ansible Community Guide
Ansible Collections Contributor Guide
ansible-core Contributors Guide
Advanced Contributor Guide
Ansible documentation style guide

/ Collection Index / Collections in the Openstack Namespace / Openstack.Cloud / openstack.cloud.server module – Create/Delete Compute Instances from OpenStack

This is the **latest** (stable) Ansible community documentation. For Red Hat Ansible Automation Platform subscriptions, see Life Cycle for version deta

openstack.cloud.server module – Create/Delete Compute Instances from OpenStack

Note

This module is part of the openstack.cloud collection (version 2.4.1).

You might already have this collection installed if you are using the `ansible` package. It is not included in `ansible-core`. To check whether it is install `ansible-galaxy collection list`.

To install it, use: `ansible-galaxy collection install openstack.cloud`. You need further requirements to be able to use this module, see Requirements f details.

To use it in a playbook, specify: `openstack.cloud.server`.

- Synopsis
- Requirements
- Parameters
- Notes
- Examples
- Return Values

Synopsis

- Create or Remove compute instances from OpenStack.

Figure 5-32 Ansible openstack.cloud.server module[1]

Note: Detailed information about using Ansible for Automation on IBM Power Systems can be found in *Using Ansible for Automation in IBM Power Environments*, SG24-8551.

5.4.1 An Ansible Execution Environment Image

We know the components of an Ansible Execution Environment Image as shown in Figure 5-33. This has the benefit of having an immutable image that can be pulled from the Ansible Hub and run on the Ansible Controller to use OpenStack APIs available in the PowerVC Cloud.

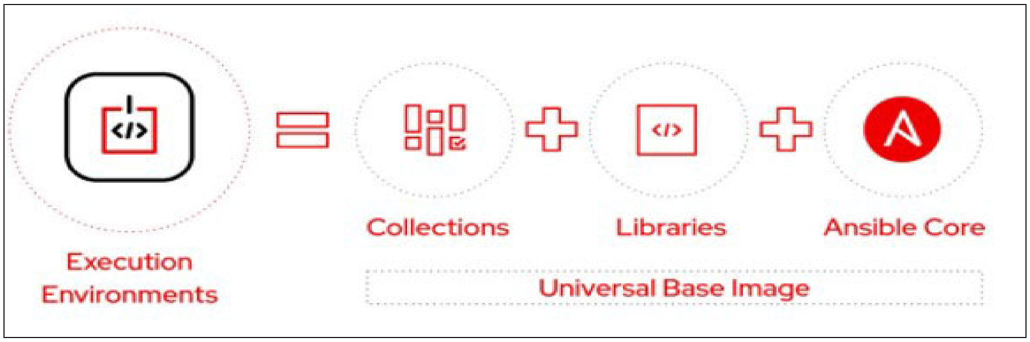


Figure 5-33 Components of an Execution Environment Image

The following sections will demonstrate how we can automate Red Hat Enterprise Linux VM tasks using Ansible Automation, ansible-navigator, on the IBM Power managed System using PowerVC as the OpenStack Cloud.

[1] https://docs.ansible.com/ansible/latest/collections/openstack/cloud/server_module.html

5.4.2 Create a collection

This section shows how we create a collection that we can use with PowerVC running on IBM Power.

1. We start with some roles that we already have as shown in Example 5-2.

Example 5-2 View roles 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. For this example using the *powervcvm* role we call the *os_server* module to create a VM in PowerVC from the playbook *roles/powervcvm/tasks/powervc.yml*.

   The Ansible galaxy *os_server* module can be used to interact directly with the PowerVC OpenStack cloud, just like any other OpenStack cloud, and pass values as shown in Example 5-3.

Example 5-3 Create new VM using PowerVC

```
 - name: Create a new 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 ]
```

There is also a play for stopping and starting lpars on IBM Power using the HMC Ansible collection.

3. In Example 5-4 we pass a server name as a variable, server_name, and search for it's details in the OpenStack PowerVC cloud namespace. The OpenStack details for the namespace are in a JSON dictionary object named *servers*.

Example 5-4 Get 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 play by *roles/powervcvm/tasks/powervc.yml*. This is automatically inherited by the following Ansible plays. This is referenced in Example 5-5.

Example 5-5 Retrieve 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
```

The inputs to the role are passed from Ansible as variables defined in a *groups_vars* directory, or any other preferred input method.

4. Create a collection named *modern.powervc_ocp*, for all the roles, in the *./collections* directory as shown in Example 5-6.

Example 5-6 Create 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 Copy roles into 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 collection *runtime.yml* with the Ansible version requirement as shown in Example 5-8.

Example 5-8 Updating collection

```
$vim modern/powervc_ocp/meta/runtime.yml

add the below to theend of the file

requires_ansible: '>=2.9.10'
```

7. To be able to import the collection into the Ansible Hub, update the collections *galaxy.yaml* to define the galaxy collection pre-requisites that are needed to run the plays in the collection as shown in Example 5-9.

Example 5-9 Add dependencies

```
$ vim modern/powervc_ocp/galaxy.yml

Add the below 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 Build the collection.

In this section we build the collection.

1. First run the collection build as shown in Example 5-10.

Example 5-10 Build 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.

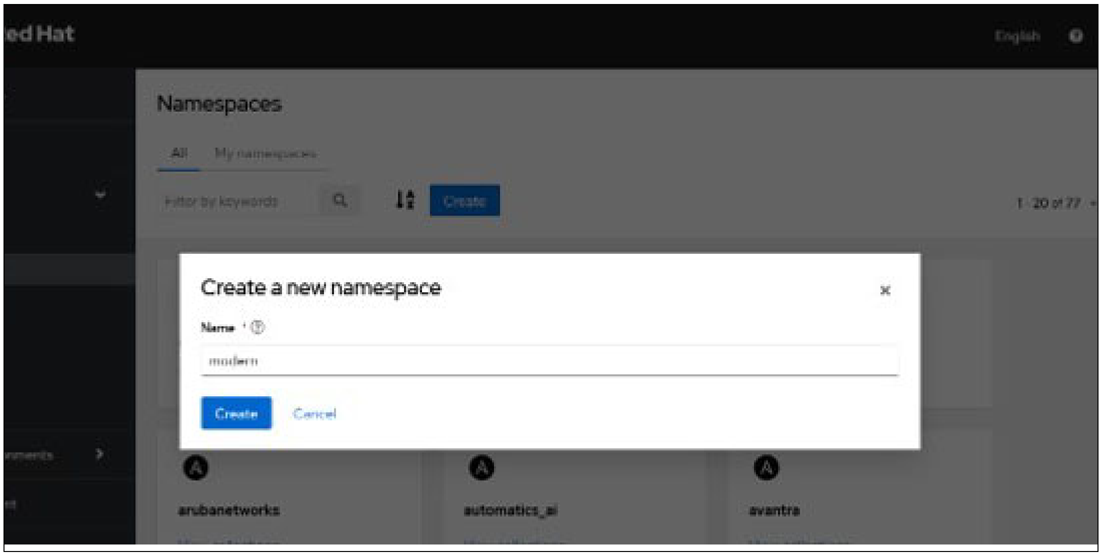
Example 5-11 Copy collection file

```
[ansi01@controller powervc_ocp]$ cp modern-powervc_ocp-1.0.0.tar.gz ../../../
[ansi01@controller powervc_ocp]$ cd ../../../
```

5.4.4 Upload the Collection to the Ansible Hub

Next we upload the collection to Ansible Hub.

1. Create the modern namespace on your Ansible Hub as shown in Figure 5-34.



Create a new namespace

Name * modern

Figure 5-34 Create new namespace

2. Ensure you have the correct key for your Automation Hub in your *ansible.cfg*. This is shown in Example 5-12.

Example 5-12 Validate the key

```
[defaults]
inventory = ./inventory/inventory
remote_user = ansi01
deprecation_warnings = false
log_path= ./ansible.log
forks = 20
collections_path = ./collections
```

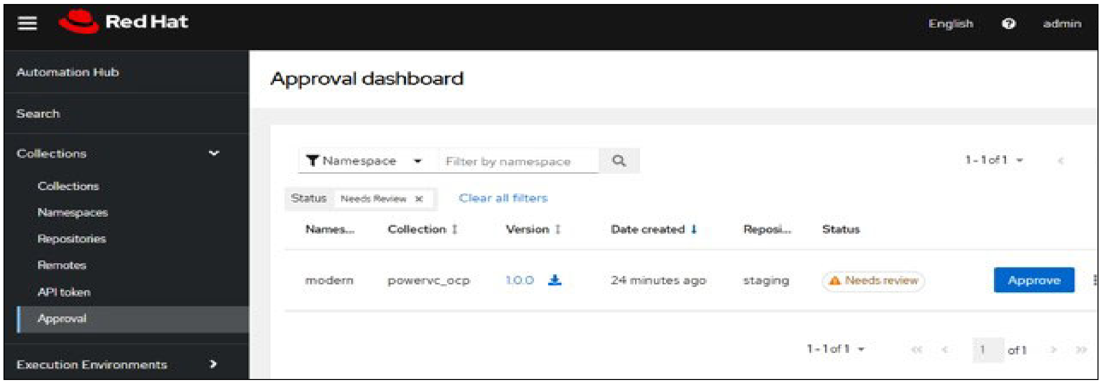
```
[galaxy]
server_list = published, rh-certified, galaxy

[galaxy_server.published]
url=https://xx.xx.xx.xx/api/galaxy/
token=xxxxxxxxxxxxxxxxxxxxxxxxxx
```

3. Now upload the collection for use and availability as shown in Example 5-13.

Example 5-13 Upload collection

```
$ ansible-galaxy collection publish modern-powervc_ocp-1.0.0.tar.gz --ignore-certs
```

4. Approve the collection as shown in Figure 5-35.



Names...	Collection	Version	Date created	Reposi...	Status
modern	powervc_ocp	1.0.0	24 minutes ago	staging	Needs review

Figure 5-35 Approve the collection

We now have in the Ansible Automation Hub a custom collection we can use in an Execution Environment Image (EEI). This is shown in Figure 5-36.

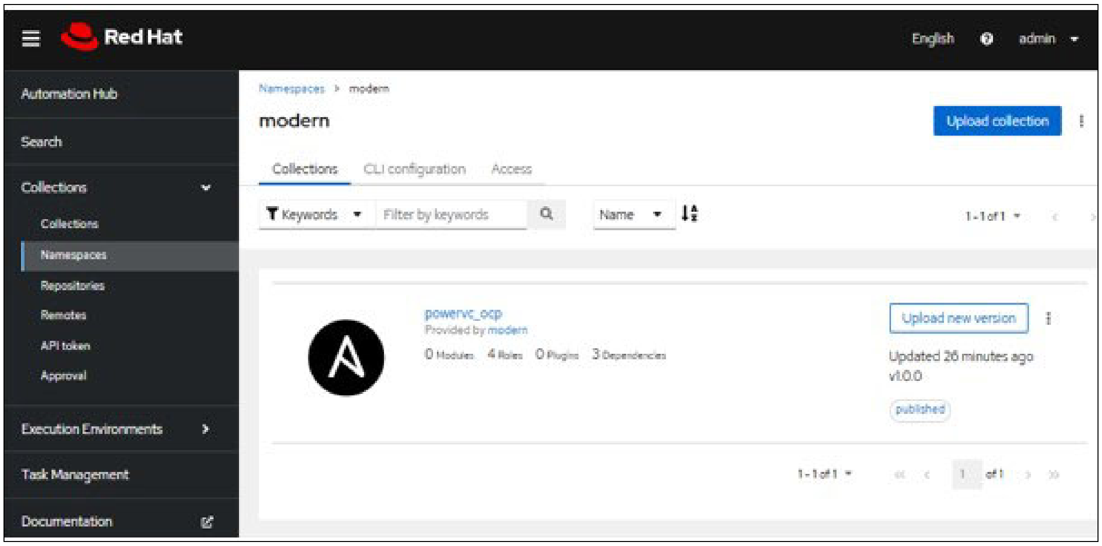


Figure 5-36 Custom collection

These custom roles are now available for any department in the organization as an Ansible Galaxy collection to use in their administration. Their Ansible workstation or Ansible controller must have the pre-requisite collections declared in the role dependencies as shown in Example 5-14 on page 135.

Example 5-14 Role dependencies

```
dependencies:
  ansible.posix: '>=1.0.0'
  openstack.cloud: '>=1.0.0'
  community.general: '>=1.1.0'
```

You should have received some warning messages when uploading the collection confirming this as shown in Example 5-15.

Example 5-15 Import 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 has been 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/
has completed
[WARNING]: Galaxy import warning message: No changelog 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
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
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 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 incorrect module path.
Collection has been successfully published and imported to the Galaxy server published
https://000.000.000.000/api/galaxy/
```

To avoid having departments maintain a workstation that may contain conflicting collections for many different requirements, we create an Ansible Execution Environment Image that includes this collection and the required collection pre-requisites. This Ansible Execution Environment Image can then run anywhere as a container.

5.4.5 Create the custom Execution Environment image file

In this section we create our Execution Environment image file.

1. Log into podman for your Ansible automation hub.
2. Create a directory to work in, e.g. **`mkdir ee-os-sdk`**
3. Create the *execution-environment.yml* for ansible-builder as shown in Example 5-16.

Example 5-16 Create 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 version: 1 of execution-environment.yml used with ansible-builder version 1 for simplicity.

Please explore details on ansible-builder version 3 for more advanced options.
https://www.redhat.com/en/blog/unlocking-efficiency-harnessing-the-capabilities-of-ansible-builder-3.0

4. Create the requirements.yml for the collections dependencies.

   Note that we have added the newly created *modern.powervc_ocp* collection and the existing *ibm.power_hmc* collection.

Example 5-17 Create 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* for Python as shown in Example 5-18.

Example 5-18 Create requirements.txt

```
# cat requirements.txt
openstackclient
openstacksdk
```

6. Create the *bindeps.txt* for the operating system.

Example 5-19 Create bindeps.txt

```
# cat bindeps.txt
libxml2-devel
libxslt-devel
python3-devel
gcc
python3-lxml
```

7. Copy the Red Hat subscription details of the local VM to two directories and make the *rhsm-auths.tar* tar file of the subscription details.

Note: This step is no longer required in version 3 of ansible-builder

Example 5-20 Create 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]$
```

8. Check that you now have in your working directory the files shown in Example 5-21.

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
```

9. Build the context for the EEI.

Example 5-22   Build 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]$
```

10. Copy the *rhsm-auths.tar* into 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 in Example 5-24 on page 139 into the *context/Containerfile*. This will provide the container with the ability to install the required rpms.

Example 5-24 Copy 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 the microdnf works as expected.

Example 5-25 Change 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 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 Build the Ansible Executable Environment Image (EEI).

1. Now use podman to build the EEI. Example 5-27 gives you 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 06d3d71e4ce40539ef8ecb8fa694dbd073cd51e065d5295aa9b07e438acfcfbb

<< truncated >>

--> Using cache 06d3d71e4ce40539ef8ecb8fa694dbd073cd51e065d5295aa9b07e438acfcfbb
--> 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 requestsexceptions-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. Login to your Ansible Automation Hub with podman as shown in Example 5-28.

Example 5-28 Login 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 hub as shown in Example 5-29.

Example 5-29 Push file to Ansible 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 6dca49abefd1 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]$
```

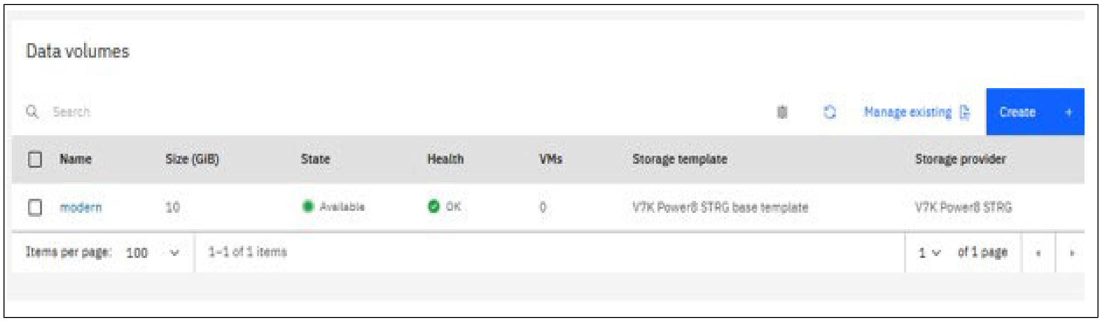
4. We can now use this EEI to run any playbooks that use Ansible OpenStack and IBM HMC modules.

5.4.7 Populate Environment Variables

The EEI has all the collections we need to run a play. All we need are the playbooks we created earlier and an *ansible.cfg*.

The plays include a disk attachment named by the environment variable *“new_vol”*.

1. Create a disk in PowerVC with any name. We shall use *modern* here as shown in Figure 5-37.



Data volumes

□ Name	Size (GiB)	State	Health	VMs	Storage template	Storage provider
□ 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 play variables in *group_vars/all/infra.yml*.

   The *site.yml* creates one VM named *vm01* and 3 worker nodes, and uses the Red Hat Enterprise Linux image from PowerVC. This is shown in Example 5-30.

Example 5-30   group_vars/all/infra.yml

```
---

# 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 needs to be populated is in *group_vars/all/os_powervc.yml*

   Enter the Cloud information for the PowerVC project you will be using here. Also, copy the PowerVC certificate `/etc/pki/tls/certs/powervc.crt` to the workstation.

   This is shown in Example 5-31.

Example 5-31 group_vars/all/os_powervc.yml

```
---

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-edbeeef2918c'
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 PowerVC images to be used.
5. Add the UUID of PowerVC flavors that can be used.

5.4.8 Use the EEI to run an Ansible Play

1. Run the *site.yml*, previously created, to build *vm01* and the worker nodes as shown in Example 5-32.

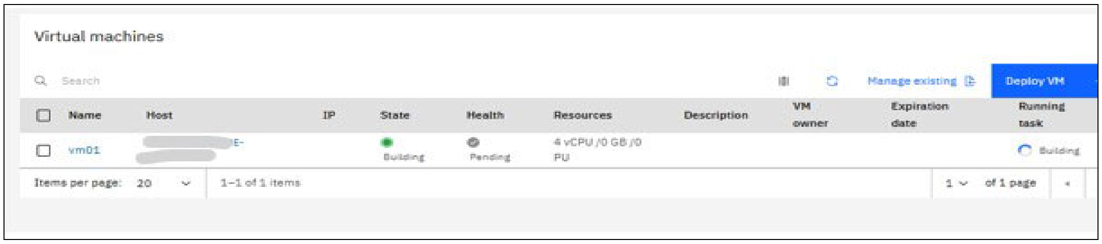
Example 5-32 Run Ansible play

```
[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 new 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. We also update the */etc/hosts* of the running container image with the new VM details to be able to update Ansible facts if needed.

   The VM is now created in PowerVC as shown in Figure 5-38.



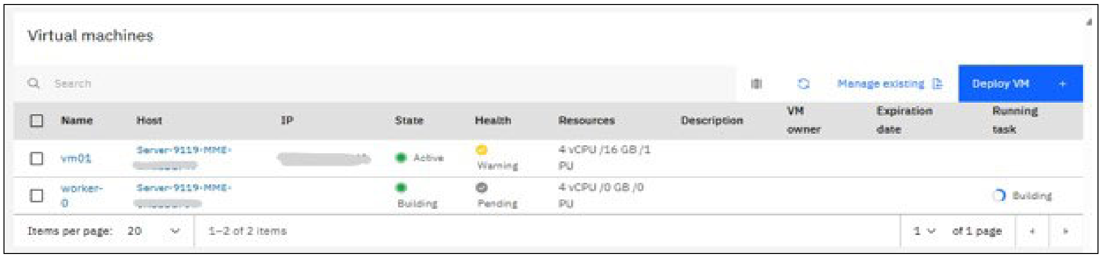
Virtual machines

□	Name	Host	IP	State	Health	Resources	Description	VM owner	Expiration date	Running task
□	vm01	[redacted]E-		● Building	Pending	4 vCPU /0 GB /0 PU				Building

Items per page: 20 1–1 of 1 items 1 of 1 page

Figure 5-38 vm01building

Likewise, the worker nodes are also building as shown in Figure 5-39.



Virtual machines

□	Name	Host	IP	State	Health	Resources	Description	VM owner	Expiration date	Running task
□	vm01	Server-9119-MME-[redacted]	[redacted]	● Active	● Warning	4 vCPU /16 GB /1 PU				
□	worker-0	Server-9119-MME-[redacted]		● Building	● Pending	4 vCPU /0 GB /0 PU				Building

Figure 5-39   Workers building

3. Normally, PowerVC allows you to create more than one VM with the same name, and adds an OpenStack UUID on the end.

   If you run this play a second time, the VMs are not created twice. A check is made in OpenStack for the VMs UUID as shown in Example 5-33.

Example 5-33   Running play again

```
[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 new 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]
*****************************************************************************************
****
```

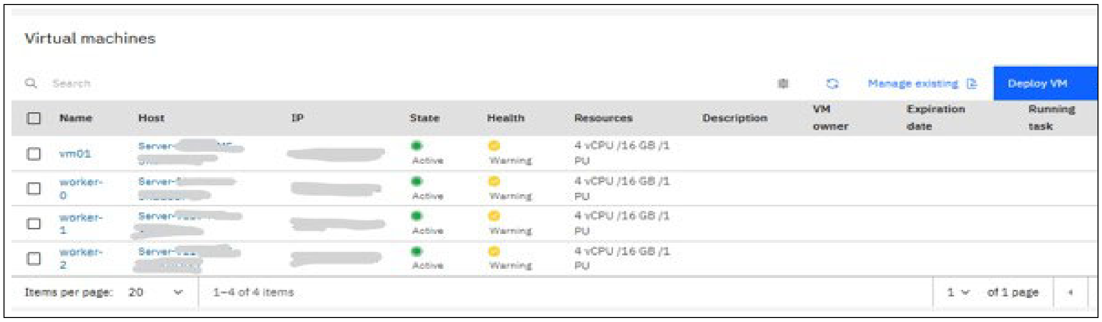
```
ok: [localhost]

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. There is a lot of information about the VM that you can access in JSON format from the PowerVC cloud. You can use this information to automate other tasks in your environment. See 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
                }
```

5. All servers are now created as seen in Figure 5-40.



Virtual machines

□	Name	Host	IP	State	Health	Resources	Description	VM owner	Expiration date	Running task
□	vm01	Server-[redacted]	[redacted]	● Active	● Warning	4 vCPU /16 GB /1 PU				
□	worker-0	Server-[redacted]	[redacted]	● Active	● Warning	4 vCPU /16 GB /1 PU				
□	worker-1	Server-[redacted]	[redacted]	● Active	● Warning	4 vCPU /16 GB /1 PU				
□	worker-2	Server-[redacted]	[redacted]	● Active	● Warning	4 vCPU /16 GB /1 PU				

Items per page: 20 1–4 of 4 items 1 of 1 page

Figure 5-40 All servers created

5.5 Automation tools

Ansible and Terraform are both powerful Infrastructure as Code (IaC) tools, but they serve different purposes and complement each other effectively. Terraform is designed to provision and manage infrastructure resources across various cloud providers (IBM Cloud, AWS, Azure, and GCP for example) 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 well together in a pipeline, where Terraform creates the necessary infrastructure resources, such as virtual machines, networks, and databases and then once the infrastructure is provisioned, Ansible takes over to configure the software and settings on those resources. Together they provide a comprehensive solution for managing both the infrastructure and the software running on it automating the entire process of provisioning and configuring infrastructure, reducing manual effort and errors.

The automation ensures that infrastructure and software are consistently deployed and configured making it easier to scale infrastructure and applications. Each tool focuses on its specific strengths, leading to a more organized and efficient workflow. By combining their strengths, you can create a powerful and efficient infrastructure automation pipeline.

5.5.1 Ansible

Ansible is an open-source, cross-platform tool for resource provisioning automation that DevOps professionals use for continuous deployment (CD) of software code by leveraging an “IaC” approach. The Ansible automation platform has evolved to deliver sophisticated automation solutions for operators, administrators, and IT decision-makers across various technical disciplines. It is an enterprise automation solution with flourishing open-source software. It operates on several UNIX like platforms, and can manage systems like UNIX and Microsoft architectures. It comes with descriptive language for describing system settings.

Because of the broad acceptance of the Ansible platform, its open-source design, and its wide support for many devices and platforms, it is becoming a dominant tool in the market. However, it is also common to use other automation tools with Ansible to do more complex automation. For example, many companies use Ansible with Terraform to provide automatic provisioning of their infrastructure.

Ansible architecture

As shown in Figure 1-2, 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.

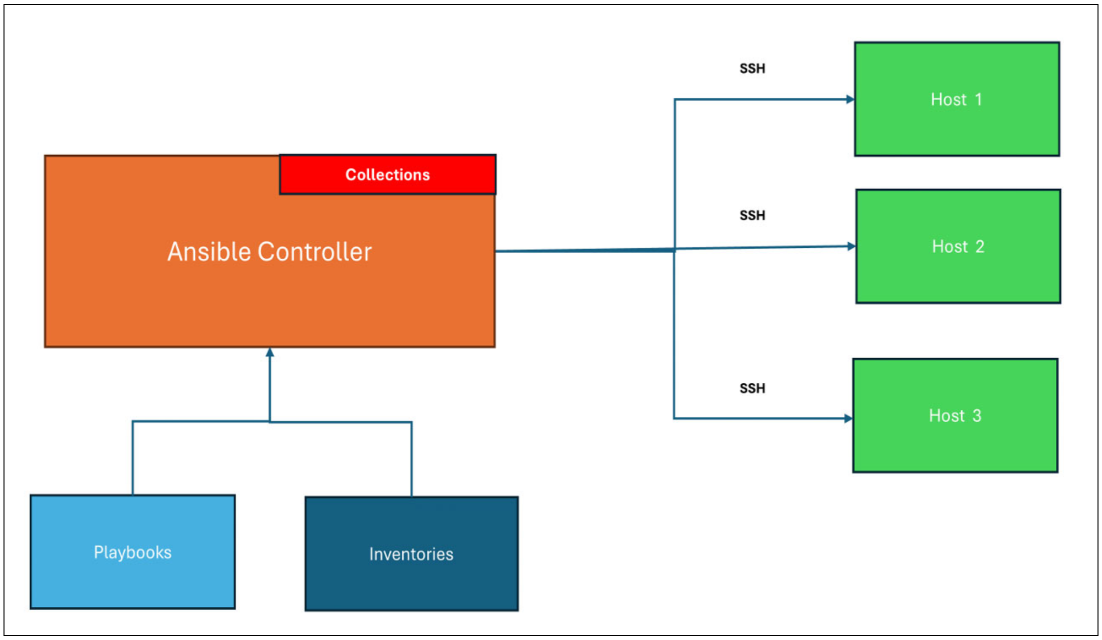


Figure 5-41 Simplified Ansible architecture

Playbooks

The heart of Ansible Automation Ansible playbooks are YAML files that define sequences of tasks to run on remote hosts. These tasks can range from installing packages to configuring services or copying files. Playbooks enable IT teams to automate infrastructure provisioning, configuration management, application deployment, and more.

Why choose Ansible

Ansible offers numerous benefits for IT professionals seeking 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 accommodate growing environments and 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 simple CLI tasks and complex workflows that are defined in playbooks.
- Extensive module library: Ansible provides a rich collection of modules for managing various systems, cloud infrastructures, and OpenStack.
- Declarative approach: With the Ansible declarative syntax, you can define the state of a system, and Ansible takes the necessary steps to achieve it.
- Ease of learning: The Ansible YAML syntax and minimal learning curve make it accessible to IT professionals at all levels.

Ansible is a powerful automation tool that can help organizations improve efficiency, scalability, and reliability in their IT infrastructure. By leveraging Ansible playbooks, IT teams can streamline routine tasks, automate complex workflows, and help ensure consistent configurations across their environments.

Options for implementing Ansible

As you decide to implement Ansible for IT management, it is essential to select the correct product and support level to meet your organization's needs. This section describes some of the options that are available to you.

Ansible Community

The community versions of Ansible primarily include the following ones:

- Ansible Core

  Ansible Core is a fundamental part of Ansible. It provides the core automation engine. It is an open-source tool that includes the basic functions for configuration management, application deployment, and task automation. Ansible Core includes modules, plug-ins, and the CLI that is needed 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. AWX provides a web-based UI, Representational State Transfer (REST) API, and task engine for managing Ansible automation at scale. AWX offers role-based access control (RBAC), job scheduling, graphical inventory management, and more. It helps users manage and scale automation efforts.

- Ansible Collections

  Ansible Collections are pre-packaged modules, roles, and plug-ins that are created and shared by the community. With Collections, users can extend Ansible functions with more content that is often maintained by the community or specific organizations. Collections can be downloaded from Ansible Galaxy, a community hub for sharing and discovering Ansible content.

- Ansible Galaxy

  Ansible Galaxy is a repository for sharing and discovering Ansible roles and collections. It is a community-driven platform where users can find reusable Ansible content to simplify automation tasks. It provides a searchable repository of roles and collections that are created by the Ansible community, which can be integrated into your automation workflows.

These community versions are suitable for individual users, small teams, and development environments but lack the formal support and advanced features that are provided by Red Hat Ansible Automation Platform.

Ansible Automation Platform

Ansible Automation Platform is a subscription-based enterprise solution that combines over 20 community projects into a fully supported automation platform. Ansible Automation Platform provides curated, certified, and validated Ansible Collections and roles from partners like IBM, Juniper, Cisco, and public cloud providers.

Here are the key considerations for choosing Ansible Automation Platform:

- Support level: Ansible Automation Platform offers enterprise-grade support, which includes SLAs for security, compatibility, and upgrades. Community options might have limited support.
- Features: Ansible Automation Platform includes features beyond Ansible Core, such as a web interface and integration with other tools.

- Cost: Ansible Automation Platform is a subscription-based product, but community options are available at no charge. Scale and complexity: For large organizations with complex automation needs, Ansible Automation Platform might be the better choice due to its enterprise-grade features and support.

By carefully evaluating these factors, you can select the Ansible offering that best aligns with your organization's goals, budget, and support requirements.

For more information on implementing automation with Ansible in an IBM Power environment refer to this IBM Redbook: *Using Ansible for Automation in IBM Power Environments*, SG24-8551

5.5.2 Terraform

Terraform is an open source tool developed by HashiCorp. It is written in the Go programming language and compiles down into an executable named Terraform. Terraform is an infrastructure as code tool that lets you build, change, and version cloud and on-premises resources safely and efficiently. Terraform provides a mechanism to access any API for any cloud provider to manage infrastructure as a service (IaaS).

Figure 5-42 shows the process involved in calling the API. The definition of which APIs to call is defined in configuration files. These configuration files are the code in that is referenced in Infrastructure as code.

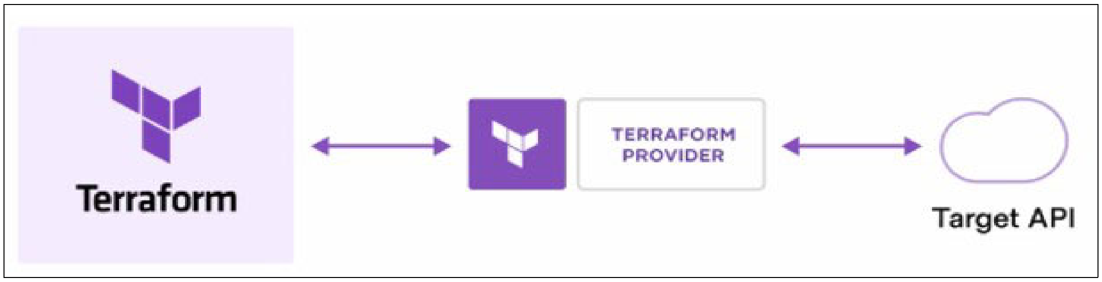
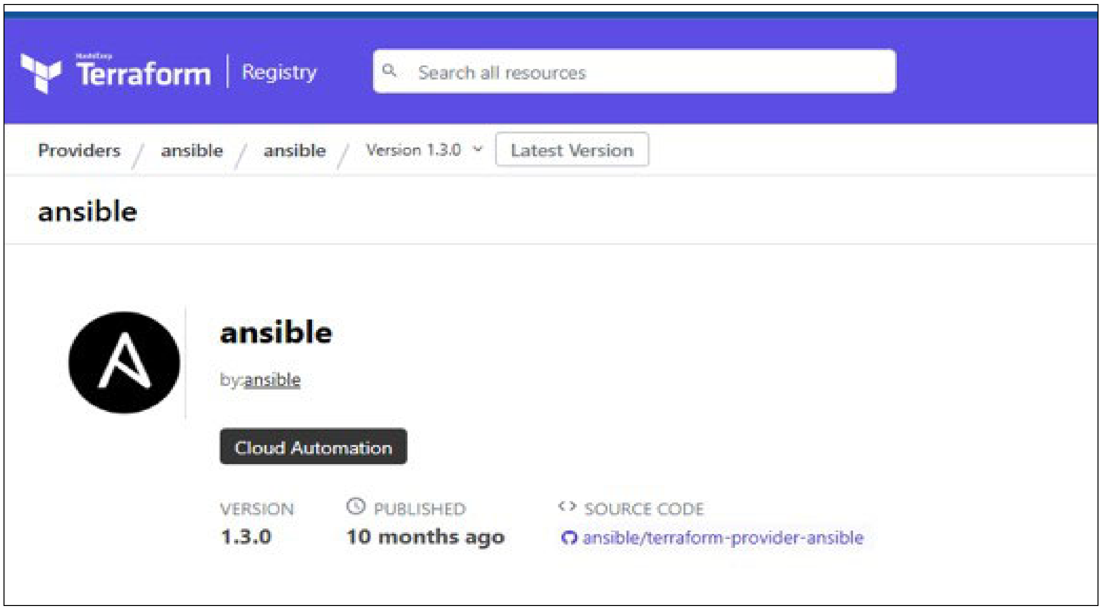


Figure 5-42 Terraform functionality

Manage any infrastructure

Terraform connects to any provider, like Ansible, to manage you infrastructure. Browse the Terraform registry for providers. The Terraform Provider for Ansible provides a more straightforward and robust means of executing Ansible automation from Terraform rather a than local-exec[2]. Figure 5-43 on page 151 shows the Ansible provider entry in the registry.

[2] https://developer.hashicorp.com/terraform/language/resources/provisioners/local-exec


Terraform | Registry
Search all resources
Providers / ansible / ansible / Version 1.3.0 ⌄ Latest Version

ansible

ansible
by:ansible

Cloud Automation

VERSION 1.3.0
PUBLISHED 10 months ago
SOURCE CODE ansible/terraform-provider-ansible

Figure 5-43 Terraform Ansible provider

The prerequisites for using the Ansible provider are as follows;

1. Install Go

   For installation instructions, refer to 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 refer to the Ansible official installation guide

Track your infrastructure

Terraform generates a plan and prompts you for your approval before modifying your infrastructure. The state of your infrastructure is kept in a file named "terraform.tfstate", which can be held in Git, Gitlab or HCP Terraform to version, encrypt, and securely share it with your team. This acts as a single source of truth for your environment.

Automate Changes

Terraform configuration files are declarative, describing the end state. So are easy to automate with tools like Ansible.

Standardize configurations

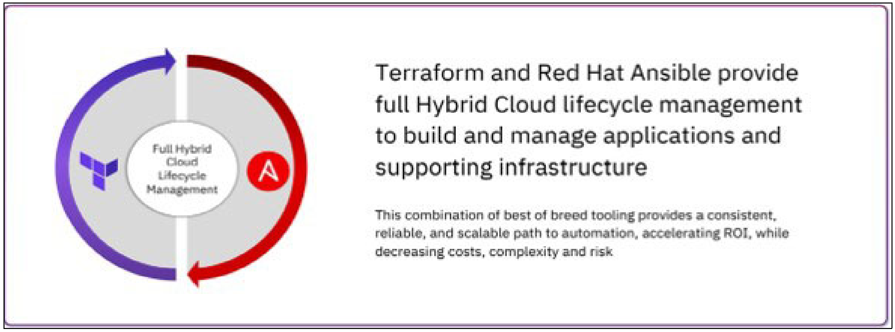
Terraform provides standardization in modules. A module consists of a collection of *.tf* and *.tf.json* files kept together in a directory. Modules are the main way to package and reuse resource configurations with Terraform.

Collaborate

Since Terraform can be distributed as configuration files and version controlled in applications like git, GitHub and HCP Terraform, these are ideal locations to share and collaborate.

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.



Terraform and Red Hat Ansible provide full Hybrid Cloud lifecycle management to build and manage applications and supporting infrastructure

This combination of best of breed tooling provides a consistent, reliable, and scalable path to automation, accelerating ROI, while decreasing costs, complexity and risk

Figure 5-44 Terraform and Ansible working together

The core Terraform workflow consists of three stages:

- Write

  You define resources, which may be across multiple cloud providers and services.

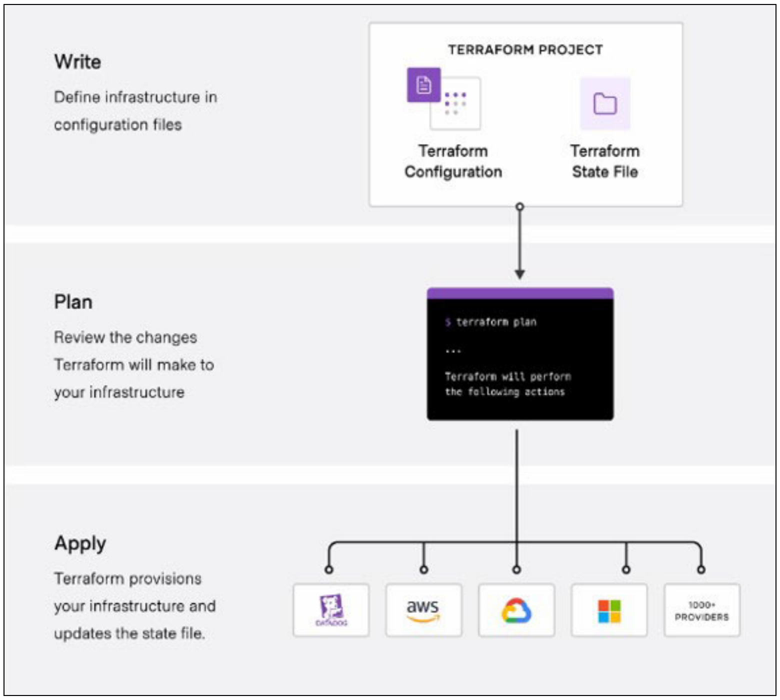
- Plan

  Terraform creates an execution plan describing the infrastructure it will create, update, or destroy based on the existing infrastructure and your configuration.

- Apply

  On approval, Terraform performs the proposed operations in the correct order, respecting any resource dependencies.

Figure 5-45 shows this process.



Write

Define infrastructure in configuration files

TERRAFORM PROJECT

Terraform Configuration

Terraform State File

Plan

Review the changes Terraform will make to your infrastructure

```
$ terraform plan
...
Terraform will perform
the following actions
```

Apply

Terraform provisions your infrastructure and updates the state file.

Figure 5-45 Terraform workflow

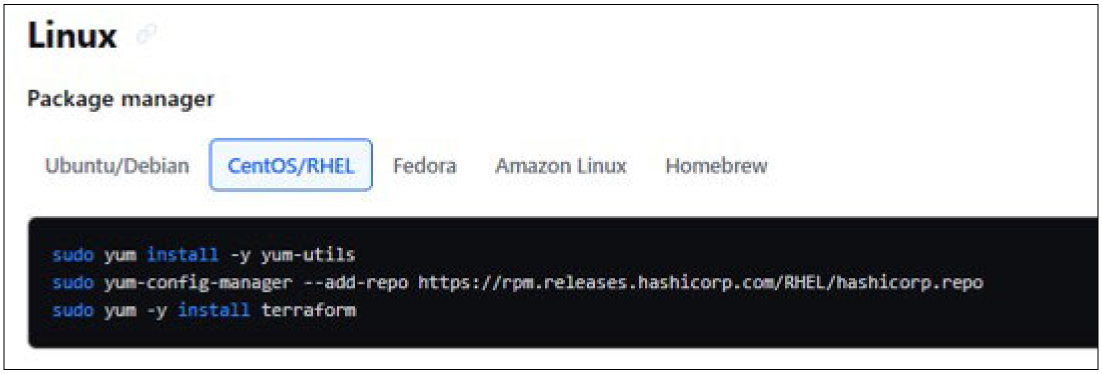
5.5.4 Terraform Plan

In our use case, we used an x86 VM for running Terraform. This could also be your Ansible Controller VM.

Install Terraform

Simply go to the Terraform site and follow instructions for your platform.

1. For RHEL on x86, you just need to add the Terraform repo and install using dnf as shown in Figure 5-46.



```
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/RHEL/hashicorp.repo
sudo yum -y install terraform
```

Figure 5-46 Installing Terraform

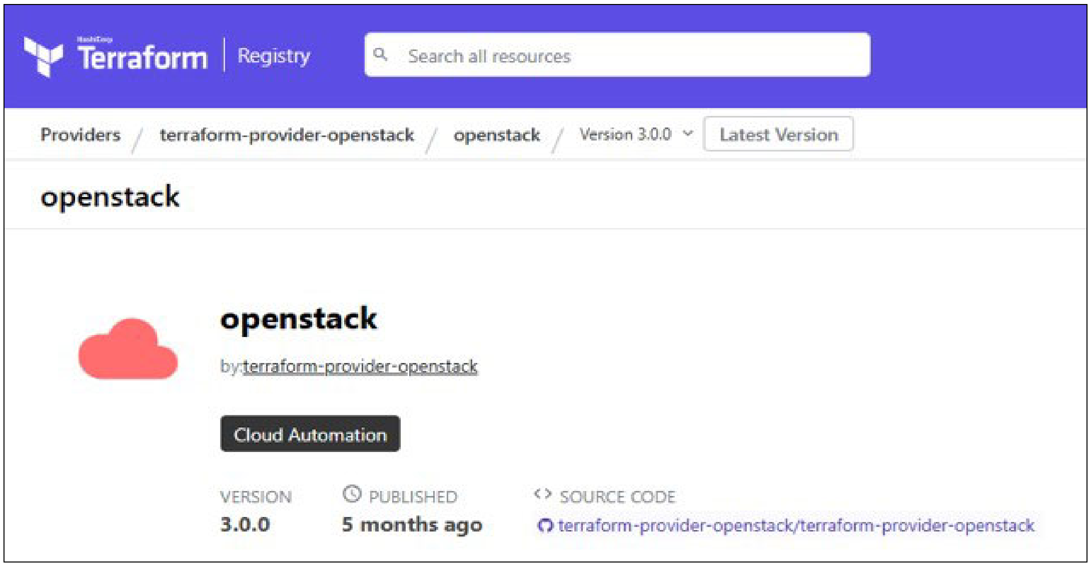
2. Then create a working directory as shown in Example 5-35.

Example 5-35 Create working directory

```
# mkdir terra-ocp
# cd terra-ocp
```

3. Define the Provider

   We are using PowerVC OpenStack so we want to use the OpenStack provider shown in Figure 5-47.



openstack

openstack

by: terraform-provider-openstack

VERSION 3.0.0

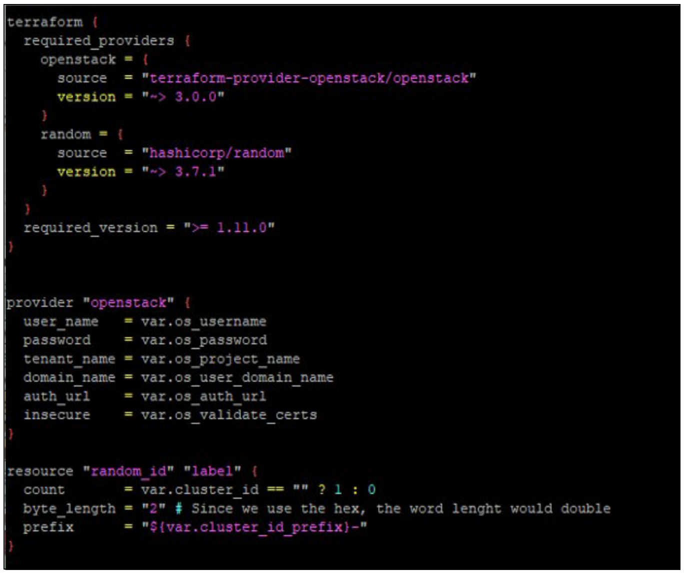
PUBLISHED 5 months ago

SOURCE CODE terraform-provider-openstack/terraform-provider-openstack

Figure 5-47 Terraform registry entry for OpenStack[3]

[3] https://registry.terraform.io/providers/terraform-provider-openstack/openstack/latest

Create a file named *providers.tf*. In this file we have identified the latest version of OpenStack, random, and Terraform from the Terraform site. This is shown in Figure 5-48.



```
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}-"
}
```

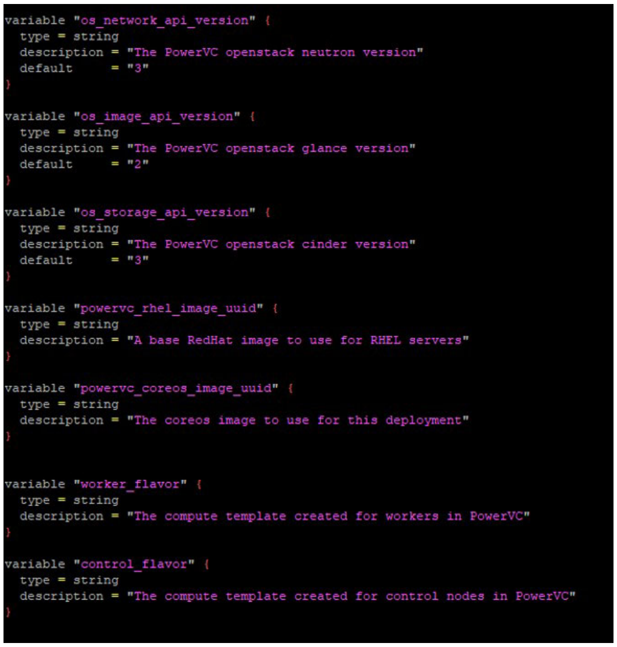
Figure 5-48 Terraform file

4. We will use random to generate a unique ID to label the lpar names and hostnames.

   The OpenStack provider uses a number of variables. These are the same OpenStack cloud values for PowerVC that we previously used for the Ansible Execution Environment plays earlier in this document. For Terraform, variables are denoted with a var prefix, and must be declared in a variables file or on the command line. For more information see this Red Hat blog – Providing Terraform with that Ansible Magic.

5. Input Variables

   We will use a file named variables.tf. In which we shall define all the variables that we will use for this Terraform plan as shown in Figure 5-49.



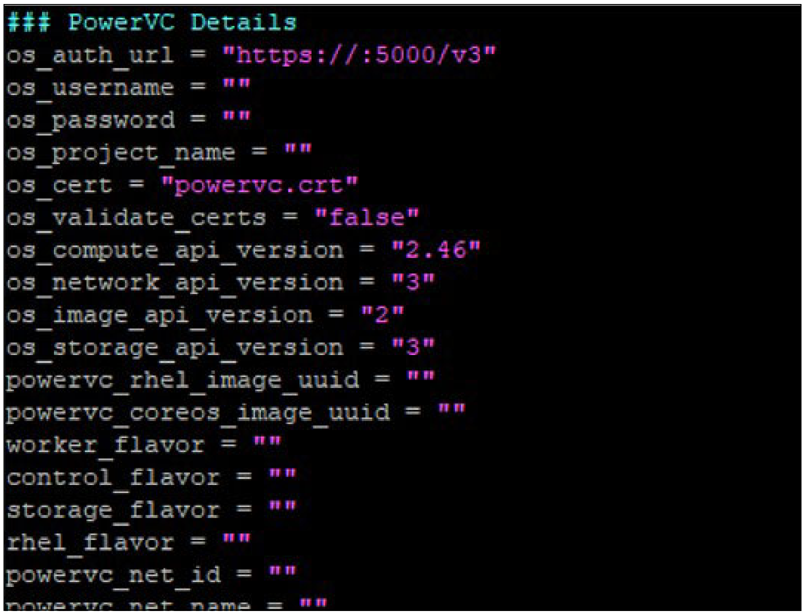
```
variable "os_network_api_version" {
  type = string
  description = "The PowerVC openstack neutron version"
  default     = "3"
}

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 variables.tf

We can define the type of variable, add a description, and a default value. If no default value is specified, Terraform will prompt you to enter the default value when you run the plan. Or you can specify the value in a variables file as input on the command line for the plan.

The var.tfvars file is used on the command line to import the values of variables which do not have a default value in variables.tf.

Figure 5-50 shows these variable definitions.



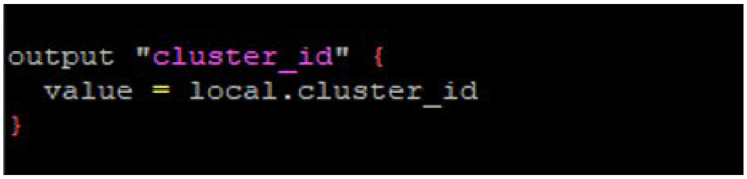
```
### PowerVC Details
os_auth_url = "https://:5000/v3"
os_username = ""
os_password = ""
os_project_name = ""
os_cert = "powervc.crt"
os_validate_certs = "false"
os_compute_api_version = "2.46"
os_network_api_version = "3"
os_image_api_version = "2"
os_storage_api_version = "3"
powervc_rhel_image_uuid = ""
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 declared so far are termed input variables.

6. Output Variables

   In an output.tf file we declare an output variable. This is a value we want to be saved that has been produced as a result of an action or execution. In this case, the variable is cluster_id as shown in Figure 5-51. To learn more about Terraform variables see https://developer.hashicorp.com/terraform/language/values.



```
output "cluster_id" {
  value = local.cluster_id
}
```

Figure 5-51 output variable

7. Locals and Modules

   Terraform will read any files in the working directory and will determine what to do with them depending on the suffix. Anything with a .tf suffix is read.

   We create a file named ocp.tf and put the action of the plan in this file. Or definition of the actions, because we will use two methods.

   a. Locals

      A local value assigns a name to an expression, so you can use the name multiple times within a module instead of repeating the expression.

This function in our ocp.tf file is the part where the cluster_id is actually created.

Example 5-36 shows creating a local value.

Example 5-36 Creating 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 and/or .tf.json files kept together in a directory. Modules are the main way to package and reuse resource configurations with Terraform.

We have a directory for each module, in a directory 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
}
```

A number of variables are declared again in these module declarations, which are passed through to the directory defining the plan for the resource to be created. which in this case is a VM. The variables passed are what is needed to create a VM in PowerVC.

We have a module declared in a similar way for each of the VMs to be created. This is shown in Example 5-38.

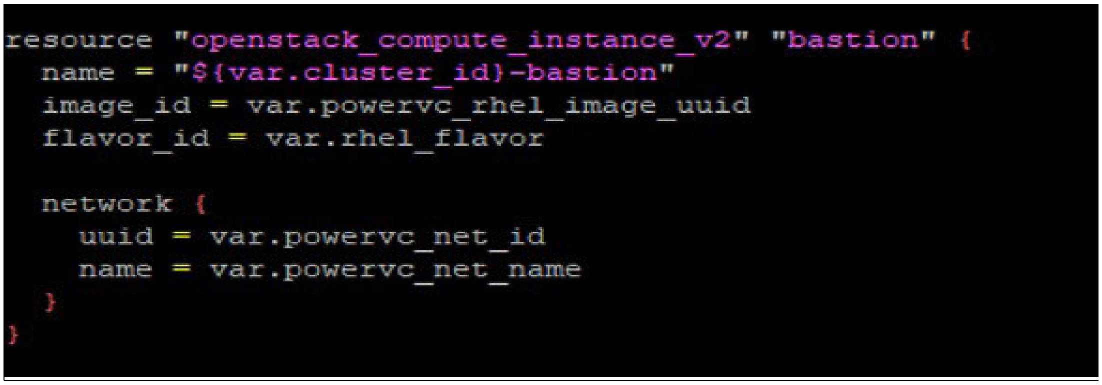
Example 5-38 List of modules

```
[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]$
```

8. Resources

The resources are the most important element in the Terraform language. Each resource block describes one or more infrastructure objects, such as virtual networks, compute instances, or higher-level components such as DNS records.

In our example there is a resource block in each module directory for each VM module definition. Figure 5-52 is the bastion resource block for creating the bastion VM which will be 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 have already been defined in the provider. So we only need to know the name of the LPAR, which image it will use, and what CPU and memory resources it will have – based on the flavor we have defined in PowerVC. Only one of the values, uuid or name, is required for the network but both values are shown for demonstration purposes, and can also be used as a double check for the instance.

The contents of the bastion module location is just a subset copy of variables required for this module, a repeat of the providers declaration, and the resource definition in the *bastion.tf* file. This is shown in Example 5-39.

Example 5-39 Bastion node directory contents

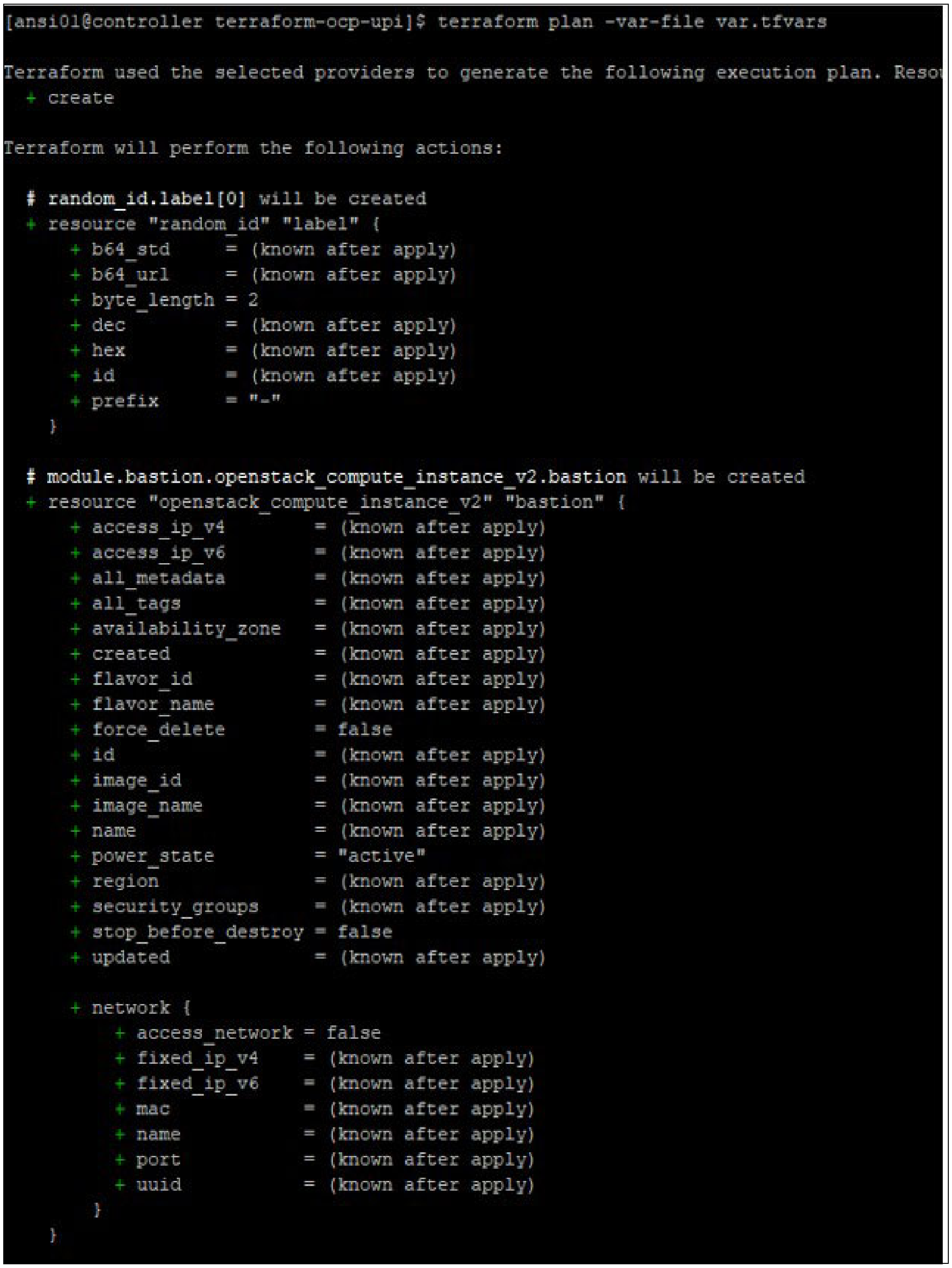
```
[ansi01@controller 1_bastion]$ pwd
/home/ansi01/git-repos/terraform-ocp-upi/modules/1_bastion
[ansi01@controller 1_bastion]$ ls -ltr
total 12
-rw-rw-r--. 1 ansi01 ansi01 247 Mar  3 18:09 bastion.tf
-rw-rw-r--. 1 ansi01 ansi01 200 Mar  4 00:42 variables.tf
-rw-rw-r--. 1 ansi01 ansi01 425 Mar  4 01:06 providers.tf
[ansi01@controller 1_bastion]$
```

9. Terraform Plan

   We can now run the command to plan, and review what will happen.

   If the plan runs successfully, then Terraform is properly configured and will running the plan will be successful.

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. Reso
  + create

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

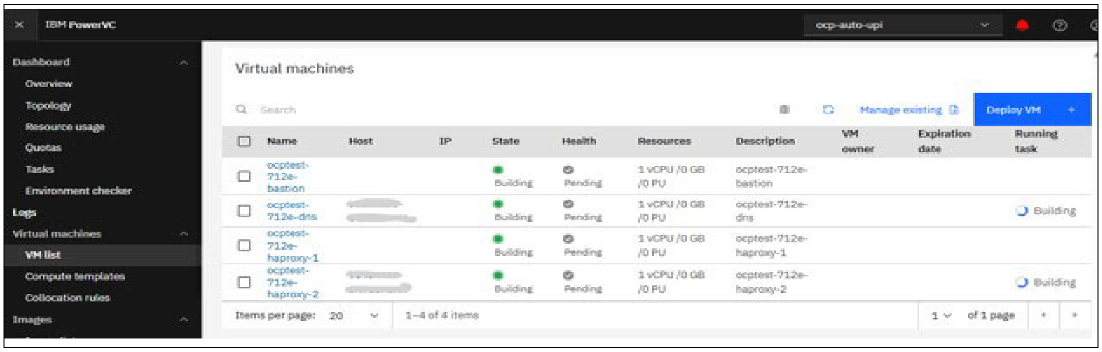
10. Terraform apply

When ready, repeat the same command with apply. Terraform will prompt you for confirmation. This is 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 described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes
```

Then sit back and watch the VMs are created in PowerVC as shown in Figure 5-54.



Virtual machines

	Name	Host	IP	State	Health	Resources	Description	VM owner	Expiration date	Running task
□	ocptest-712e-bastion			● Building	Pending	1 vCPU /0 GB /0 PU	ocptest-712e-bastion			
□	ocptest-712e-dns	[redacted]		● Building	Pending	1 vCPU /0 GB /0 PU	ocptest-712e-dns			Building
□	ocptest-712e-haproxy-1			● Building	Pending	1 vCPU /0 GB /0 PU	ocptest-712e-haproxy-1			
□	ocptest-712e-haproxy-2	[redacted]		● Building	Pending	1 vCPU /0 GB /0 PU	ocptest-712e-haproxy-2			Building

Items per page: 20 1–4 of 4 items 1 of 1 page

Figure 5-54 PowerVC GUI showing creation of LPARs

5.5.5 Configuring and customizing Terraform defined resources

At the end of the previous step, we have the VMs installed, but no applications are installed and the VMs have not been customized. To avoid having to manually set up each LPAR, further automation is required.

Using Terraform provisioners

We could complete the customization of the VM using the local-exec or remote-exec method in Terraform. It is generally recommended to minimize the use of provisioners. as configuration management tools like Ansible or cloud-init are often better suited for complex configuration tasks. In addition provisioners can make Terraform configurations less predictable and harder to maintain.

There are two distinct types of provisioner for Terraform:

- local-exec provisioner

  The local-exec provisioner invokes a local executable after a resource is created. This invokes a process on the machine running Terraform, not on the resource.

- remote-exec provisioner

  The remote-exec provisioner invokes a script on a remote resource after it is created. This can be used to run a configuration management tool, or other task like bootstrap into a cluster. The remote-exec provisioner requires a connection and supports both *ssh* and *winrm*.

Provisioners are best used for simple, one-time tasks that cannot be handled by other means. For more information on using Provisioners in Terraform see this HashiCorp Document. The next section discusses the use of Ansible instead of Terraform provisioners.

Using Ansible provider for Terraform

While it is possible to use the Terraform provisioners to customize your VMs, we recommend that you instead use Ansible to do the final customization instead. This section describes using the Ansible provider for Terraform.

Ansible Provider Pre-Requisites

To prepare for using the Ansible provider for Terraform:

- install Go

  The official installation guide can be found at https://go.dev/doc/install.

- install Terraform

  The official installation guide can be found at https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli

- install Ansible\

  The official installation guide can be found at https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html

Installing prerequisites

1. In our environment we already have Terraform and Ansible installed, so we only need to install Go. After we finish the installation, we verify the installation as shown in Example 5-41.

Example 5-41 Install Go

```
[ansi01@controller terraform-ocp-upi]$ go version
go version go1.24.0 linux/amd64
```

2. Next we download and make the teraform-provider-ansible.

   Put the provider in the same registry.terraform.io for the project as you have OpenStack. This is 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:** Details of how to download and make the terraform-ansible-provider can be found here.

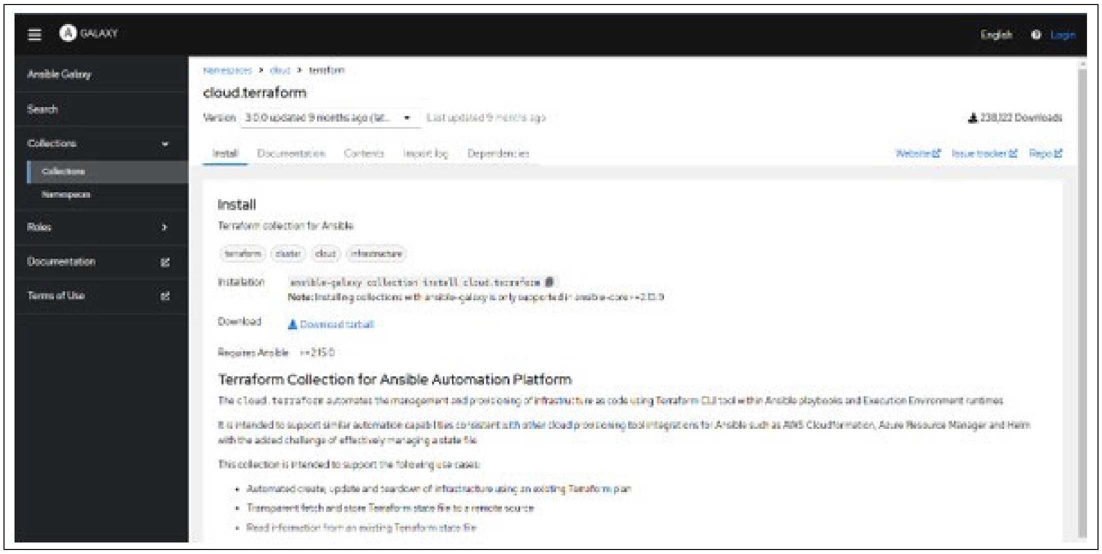
There are some example playbooks in the provider directory. But if you already have plays, and collections built into an Ansible Automation Execution Environment Image, re-working this back into simple playbooks could take some time, and will expose code on the controller.

5.5.6 Ansible Terraform Collection

Using the Ansible Collection cloud.terraform could be the answer to integrating existing EEI's for your AAP environment. In this scenario, you would be using Ansible to call Terraform as shown in Figure 5-55.

Figure 5-55 Ansible and Terraform working together

The collection is shown in Figure 5-56.



cloud.terraform

Version 3.0.0 updated 9 months ago (lat... Last updated 9 months ago

238,122 Downloads

Install

Terraform collection for Ansible

terraform cluster cloud infrastructure

Installation `ansible-galaxy collection install cloud.terraform`

Note: Installing collections with ansible-galaxy is only supported in ansible-core >=2.13.9

Download Download tarball

Requires Ansible >=2.15.0

Terraform Collection for Ansible Automation Platform

The `cloud.terraform` automates the management and provisioning of infrastructure as code using Terraform CLI tool within Ansible playbooks and Execution Environment runtimes.

It is intended to support similar automation capabilities consistent with other cloud provisioning tool integrations for Ansible such as AWS Cloudformation, Azure Resource Manager and Helm with the added challenge of effectively managing a state file.

This collection is intended to support the following use cases:

- Automated create, update and teardown of infrastructure using an existing Terraform plan
- Transparent fetch and store Terraform state file to a remote source
- Read information from an existing Terraform state file

Figure 5-56 Ansible Terraform collection

Integrating Ansible with Terraform allows for seamless automation of infrastructure provisioning and configuration management. Terraform is used to define and create the infrastructure, such as virtual machines or cloud resources, while Ansible configures and deploys software on these resources.

5.6 Power Enterprise Pools & Cloud Management Console

IBM Power Enterprise Pools is a technology for dynamically sharing processor and memory activations and OS licenses among a group of IBM Power Systems. The solution is ideal for further improving the flexibility, load balancing, and disaster recovery planning of IBM Power Systems. This technology enables capability to clients deploying and managing a private cloud infrastructure.

5.6.1 Power Enterprise Pools 1.0

PEP 1.0 is collection of CPU and memory mobile activations which can be manually reallocated among the servers within the Power Enterprise Pool. Enterprise Pool 1.0 is monitored and managed from the Hardware Management Console.

5.6.2 Power Enterprise Pools 2.0

IBM Power Private Cloud with Shared Utility Capacity (also known as Power Enterprise Pools 2 or PEP2) offers enterprises 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 Systems, optimizing utilization 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.

Power Enterprise Pools 2.0 (PEP2) enables the pooling of CPU and memory resources across a defined set of IBM Power servers, offering greater flexibility in how these resources are utilized.

Each server in the pool is ordered with a number of Base Processor and Memory activations. When a PEP2 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 can be accessed seamlessly and automatically as demand arises. If resource usage exceeds the aggregated base capacity, the excess is measured in real time and billed as metered capacity, either deducted from Prepaid Capacity Credits or invoiced on a monthly basis.

The resources used within PEP 2.0 are tracked and monitored by IBM Cloud Management Console (CMC).

5.6.3 Cloud Management Console

The IBM Cloud Management Console for Power Systems provides a consolidated view of the Power Systems cloud landscape, no matter how many systems or data centers comprise it, including inventory of systems and virtual components, consolidated performance data to optimize utilization and performance across all your data centers, and aggregated logging information to provide additional insights.

CMC runs as a service and it is hosted in the IBM Cloud. It can be accessed securely at any time enabling system administrators to easily run reports and gain insight into their Power cloud deployments. As private and hybrid cloud deployments grow, enterprises need new insight into these environments. Tools that provide consolidated information and analytics can be key enablers to smooth operation of infrastructure.

Figure 5-57 shows the interaction of the CMC as it monitors the servers within the different pools.

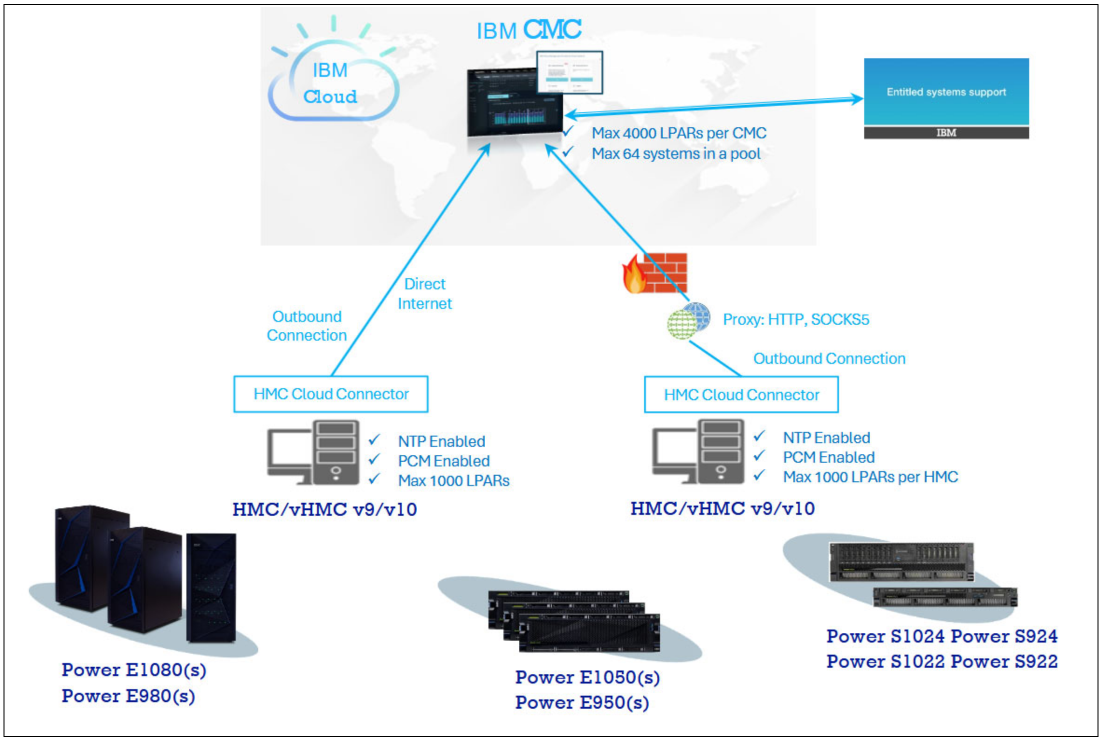


Figure 5-57 IBM Cloud Management Console overview

For more information on Power Enterprise Pools and the Cloud Management Console 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 extensive and powerful networking tools and technologies, which you can use to enable more flexibility, better security, and enhanced usage of hardware resources. 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:

- Virtual network

  Enables inter-partition communication without assigning a physical network adapter to each partition. If the virtual network is bridged, partitions can communicate with external networks. A virtual network is defined by its name or VLAN ID and the associated virtual switch. Use the HMC wizard to configure a virtual network.

- Virtual ethernet adapter

  Enables a client partition to send and receive network traffic without a physical ethernet adapter.

- Virtual switch

  An in-memory, hypervisor implementation of a layer-2switch. The Power Hypervisor implements a virtual ethernet switch with the IEEE 802.1Q capabilities for virtual local area networking (VLAN) on an IEEE 802.3 Ethernet network. The standard defines a system of VLAN tagging for Ethernet frames and the accompanying procedures to be used by bridges and switches in handling such frames.

- Virtual network bridge

  A software adapter that bridges physical and virtual networks to enable communication. A network bridge can be configured for failover or load sharing. You cannot configure a virtual network bridge with more than two trunk adapters by using the HMC Graphical User Interface (GUI) or the HMC REST API.

  However, if required, you can use the HMC command-line interface (CLI) and VIOS commands to create more than two trunk adapters, with the same VLAN configuration. Additionally, you can set different priorities for trunk adapters across Virtual I/O Servers.

  After creating trunk adapters, the HMC REST API or the HMC GUI does not support any operation on that virtual network bridge. You must use the HMC CLI and VIOS commands to delete any trunk adapters and continue the operation on that virtual network bridge by using the HMC REST API or the HMC GUI.

- Link aggregation device.

  A link aggregation (also known as Etherchannel) device is a network port-aggregation technology that provides several physical Ethernet adapter ports in a single aggregated device. Link aggregation may follow the Link Aggregation Control Protocol (LACP) for Ethernet defined in IEEE 802 1AX or the previous IEEE 802.3ad. Link aggregation increases the bandwidth and resilience of ethernet connections, providing more throughput over a single IP address.

> **Note:** For an overview of PowerVM networking concepts please visit
> `https://www.ibm.com/docs/en/power10?topic=mvn-powervm-networking-concepts`

5.7.1 Shared Ethernet Adapter technology

PowerVM introduced the Shared Ethernet Adapter (SEA) in order to provide the sharing of Ethernet adapters across multiple LPARs. This provides the flexibility most customers require, but the existing SEA-based virtual networking solution incurs layered software overhead and multiple data copies from the time a packet is committed for transmission on the VEA to the time the packet is queued on the physical NIC for transmission (same issues apply for receive packets). This can create performance issues when a large amount of data is being transmitted over the SEA.

Performance Considerations

Here are some performance and administrative considerations when using the Shared Ethernet Adapter.

- The range of port vlanIDs to be consumed and considered “throw away” vlanIDs when creating tagged networks can be large when a configuration of one tagged vlanID to one virtual ethernet adapter is implemented. Work with the network teams to determine a range that would never be used in the future, or reserved for PowerVM networking.
- Implement the increase of network buffers for the virtual ethernet adapter, and isolate high performance workload vlan IDs on their own virtual ethernet adapter to maximize this buffer allocation for the workload.
- The Shared Ethernet Adapter is a Network Bridge, and some tuning is required in the VIOS and on the client VMs to achieve the speed or throughput required for modern 10 Gb/s, 25 Gb/s networks.

Figure 5-58 illustrates network adapter sharing across the various LPARs.

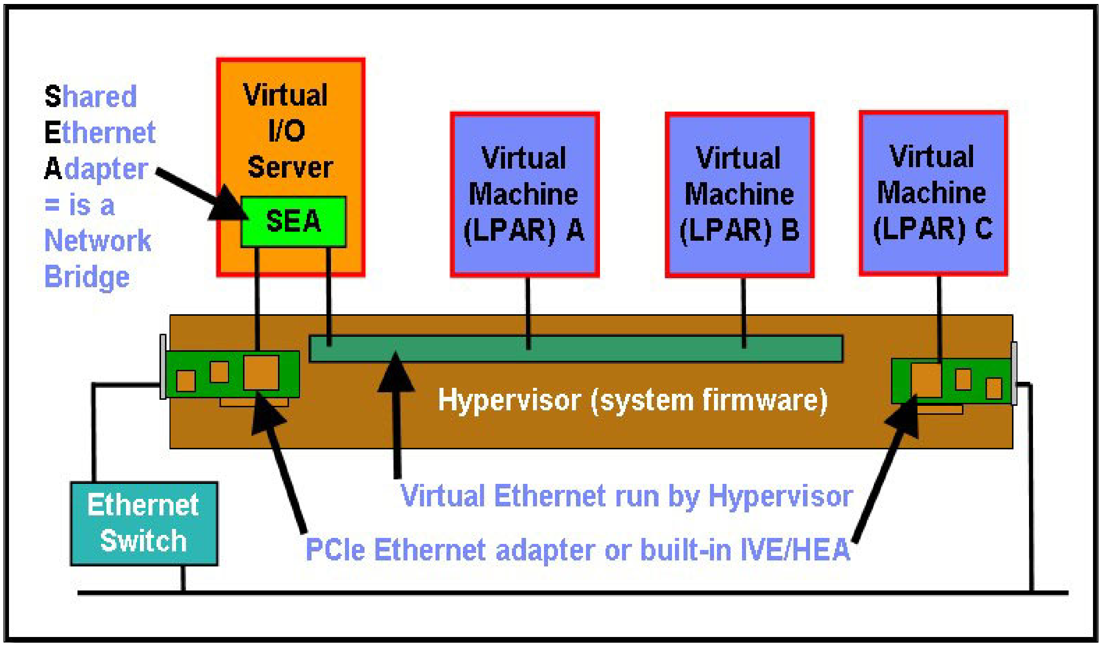


Figure 5-58 SEA as a network bridge[4]

5.7.2 Virtual Network Interface technology

SR-IOV is an extension to the Peripheral Component Interconnect® (PCI) Express specification to allow multiple partitions that are running simultaneously within a single system to share a PCI Express device. With the introduction of the SR-IOV capable adapters, the industry created a method of sharing a network adapter in a PCIe slot with multiple partitions. Previous to this technology, a PCIe slot was dedicated to a single partition and only could be shared through the use of a virtual I/O server.

An SR-IOV-capable adapter can either be assigned to a partition in dedicated mode or be owned by the hypervisor when switched to shared mode. In shared mode, the adapter can be utilized by multiple logical partitions simultaneously. This allows a single adapter with multiple physical ports to deliver high-performance networking or data processing capabilities to several partitions through logical ports, enabling efficient resource sharing while maintaining performance.

SR-IOV technology

An SR-IOV adapter allows the creation of multiple virtual replicas of a PCI function, called a Virtual Function (VF), and each VF can be assigned to an LPAR independently. The SR-IOV VF operates with little software intervention providing superior performance with very little CPU overhead. An SR-IOV virtual function is a PCIe function defined by the Single Root I/O Virtualization and Sharing specification. The VF is the connection for an SR-IOV logical port.

SR-IOV logical port

An SR-IOV logical port is an I/O device created for a partition or a partition profile using the management console (HMC) when a user intends for the partition to access an SR-IOV adapter Virtual Function.

[4] https://www.ibm.com/support/pages/powervm-virtual-ethernet-speed-often-confused-vios-sea-speed

vNiC

A vNIC (Virtual Network Interface Controller) is a new PowerVM virtual networking technology that delivers enterprise capabilities and simplifies network management when utilizing SR-IOV adapters. It is a high performance technology that when combined with an SR-IOV Network Interface Controller (NIC) provides bandwidth control and Quality of Service (QoS) capabilities at the virtual NIC level. The vNIC technology significantly reduces virtualization overhead resulting in lower latencies and reducing the server resources (CPU, memory) required for network virtualization.

With the introduction of SR-IOV capable adapters, it is now possible to share a single Ethernet adapter in a more efficient method, reducing some of the performance issues that were experienced with the use of the SEA.

5.7.3 SR-IOV and SEA comparison

A comparison of SEA and vNIC data flows can be seen in Figure 5-59. Note the differences between the two flows in that in the vNIC implementation the control packets flow through the Power Hypervisor, but the data packets flow directly to the partition. The reduction in software and device latency can be seen from the direct path taken by packets to the client LPAR.

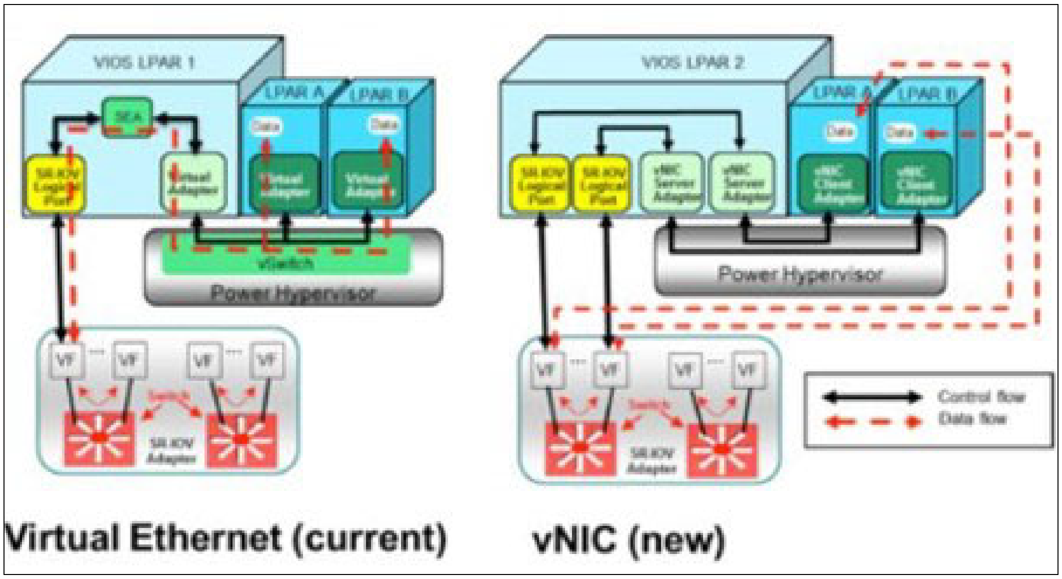


Figure 5-59 SEA data flow compared to vNIC data flow

The vNIC is a type of virtual Ethernet adapter, that is configured on the LPAR, where each vNIC is backed by an SR-IOV logical port (LP) that is available on the VIO server. This ensures that the client LPAR is eligible for Live Partition Mobility. A technology named Logical Redirected DMA (LRDMA) enables the vNIC to transmit and receive buffers directly to the remote SR-IOV logical port.

The vNIC control and data flow can be seen again in Figure 5-60.

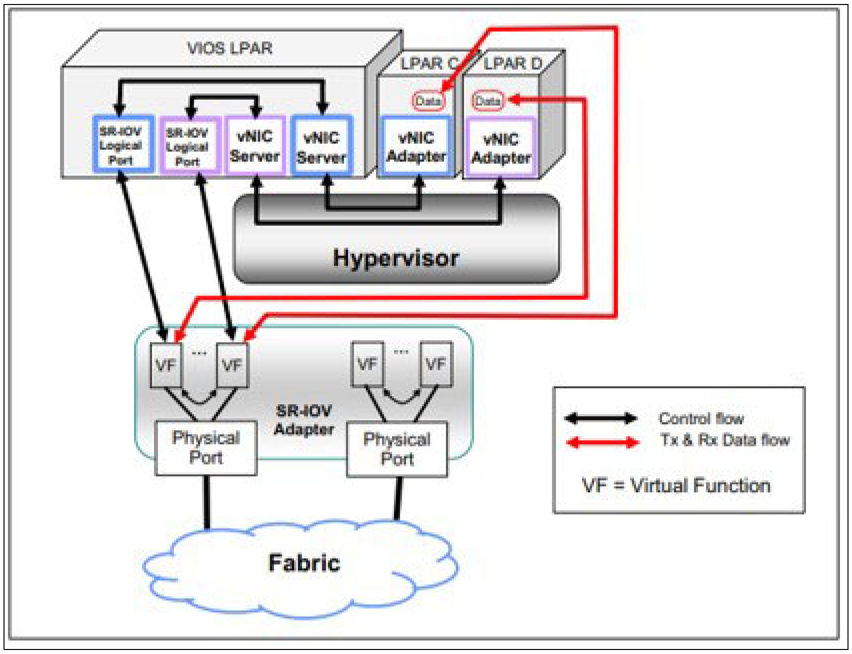


Figure 5-60 Data flow using vNIC on SR-IOV adapter

The key element of the vNIC model is a one-to-one mapping between a vNIC virtual adapter in the client LPAR and the backing SR-IOV logical port in the VIOS. With this model, packet data for transmission (similarly for receive) is moved from the client LPAR memory to the SR-IOV adapter directly without being copied to the VIOS memory. The benefits of bypassing the VIOS are reduction of the processing of a memory copy (specifically lower latency), and the reduction in the CPU and VIOS memory consumption (greater efficiency).

Table 5-1 compares and contrasts the different technologies available for network connectivity.

Table 5-1 Comparison of network technologies

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

a. SR-IOV can optionally be combined with VIOS and virtual Ethernet to use higher-level virtualization functions like Live Partition Mobility (LPM); however, client partition will not receive the performance or QoS benefit.

b. Some limitations apply. See FAQ on link aggregation.

c. Generally better performance and requires fewer system resources when compared to SEA/virtual Ethernet.

SR-IOV adapters

Not all Ethernet adapters are SR-IOV capable. Table 5-2 shows the adapters that are SR-IOV Capable when running on Power10 servers. For a list of adapters for previous generation servers see this IBM Community document.

Table 5-2 vNIC capable adapters on Power10 servers

SR-IOV Capable Network I/O Adapters	FCs	Server & Attached I/O Expansion Drawer Adapters				
		S1012	**S1022, S1022s L1022, (EMX0, ENZ0)**	**S1014, S1024, L1024, (EMX0, ENZ0)**	**E1050 (EMX0, ENZ0)**	**E1080 (EMX0, ENZ0)**
PCIe3 2-Port 10GbE NIC & RoCE SR/Cu Adapter[a]	EC2R, EC2S		EC2R (EC2S)	EC2S (EC2S)	EC2S (EC2S)	EC2R (EC2S)
PCIe3 2-Port 25/10GbE NIC & RoCE SR/Cu Adapter	EC2T, EC2U		EC2T (EC2U)	EC2U (EC2U)	EC2U (EC2U)	EC2T (EC2U)
PCIe4 2-port 100/40GbE NIC & RoCE QSFP28 Adapter x16	EC67, EC66		EC67	EC66	EC66	EC67
PCIe4 x16 2-port 100/40GbE NIC & RoCE QSFP28 Adapter[b]	EC75, EC76		EC75	EC76	EC76	EC75
PCIe4 2-Port 25/10/1 Gb RoCE SFP28 Adapter[c,d]	EC71, EC72	EC71	EC71 (EC72)	EC72 (EC72)	EC72 (EC72)	EC71 (EC72)
PCIe4 4-Port 25/10/1 GbE RoCE SFP28 Adapter[e]	EN24, EN26		(EN26)[f]	EN26 (EN26)[f]	EN26 (EN26)[f]	EN24 (EN26)[f]

a. Withdrawn
b. SR-IOV support available on Power10 Servers with FW1030
c. SR-IOV support 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 available on Power10 Servers with FW1060.10
f. Not supported in the EMX0

Each SR-IOV adapter can support different number of logical ports. Table 5-3 shows the number of logical ports/VFs supported per adapter supported on Power as well as its connectivity speed options.

Table 5-3 Number.

SR-IOV Capable Network I/O Adapters	Feature codes	Physical port link speed	# of logical ports per physical port	# of logical ports per physical port
PCIe2 4-port (2x10GbE+2x1GbE) SR Optical fiber and RJ45	EN0J, EN0H, EL38, EL56	1 Gb	4	48
		10 Gb	20	
PCIe2 4-port (2x10GbE+2x1GbE) copper twinax and RJ45	EN0L, EN0K, EL3C, EL57	1 Gb	4	48
		10 Gb	20	
PCIe2 4-port (2x10GbE+2x1GbE) LR Optical fiber and RJ45	EN0N, EN0M	1 Gb	4	48
		10 Gb	20	
PCIe3 4-port 10GbE SR optical fiber	EN16, EN15	10 Gb	16	64
PCIe3 4-port 10GbE copper twinax	EN18, EN17	10 Gb	16	64
PCIe3 LP 2-Port 10Gb NIC&ROCE SR/Cu Adapter	EC2R, EC2S	10 Gb	40	80

SR-IOV Capable Network I/O Adapters	Feature codes	Physical port link speed	# of logical ports per physical port	# of logical ports per physical port
PCIe3 LP 2-Port 25/10Gb NIC&ROCE SR/Cu Adapter	EC2T, EC2U	25/10 Gb	40	80
PCIe3 LP 2-port 100GbE NIC&RoCE QSFP28 Adapter x16	EC3L, EC3M	40/100 Gb	60	120
PCIe4 LP 2-port 100GbE NIC & RoCE QSFP28 Adapter x16	EC66, EC67	40/100Gb	60	120
PCIe4 x16 2-port 100/40GbE NIC & RoCE QSFP28 Adapter	EC75, EC76	40/100 Gb	80	160
PCIe4 2-Port 25/10/1 Gb RoCE SFP28 Adapter[a]	EC71, EC72	1/10/25 Gb	40	80
PCIe4 4-Port 25/10/1 GbE RoCE SFP28 Adapter	EN24, EN26	1/10/25 Gb	20	80

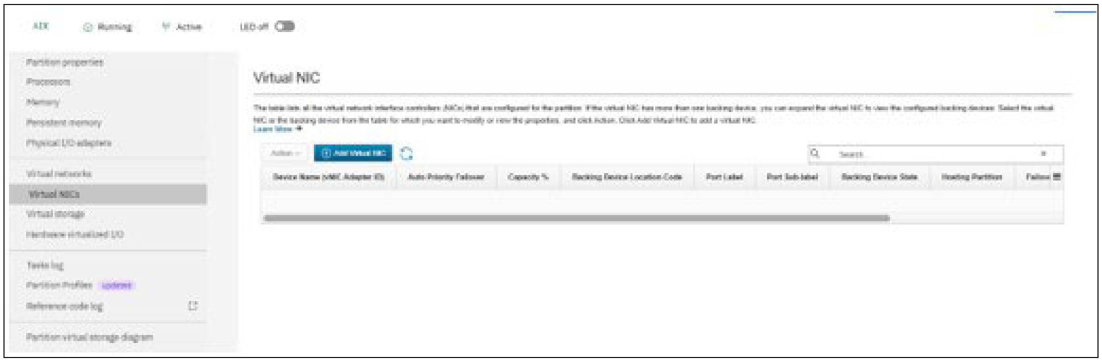
[a] If the physical port is configured for a 1Gb link speed the number of logical ports per physical port should be limited to a maximum of 4.

Restriction: The maximum number of SR-IOV shared mode enabled adapters per system is 32.

5.7.4 vNIC Configuration

The vNIC configuration happens on the HMC in a single step (only in the Enhanced GUI). When adding a vNIC adapter to the LPAR, it will create all necessary adapters automatically on the VIOS (SR-IOV logical port, vNIC server adapter) and on the LPAR (vNIC client adapter). No additional manual configuration is needed at VIOS side.

In your HMC you can access the LPAR details and then Virtual NICs (see Figure 5-61).

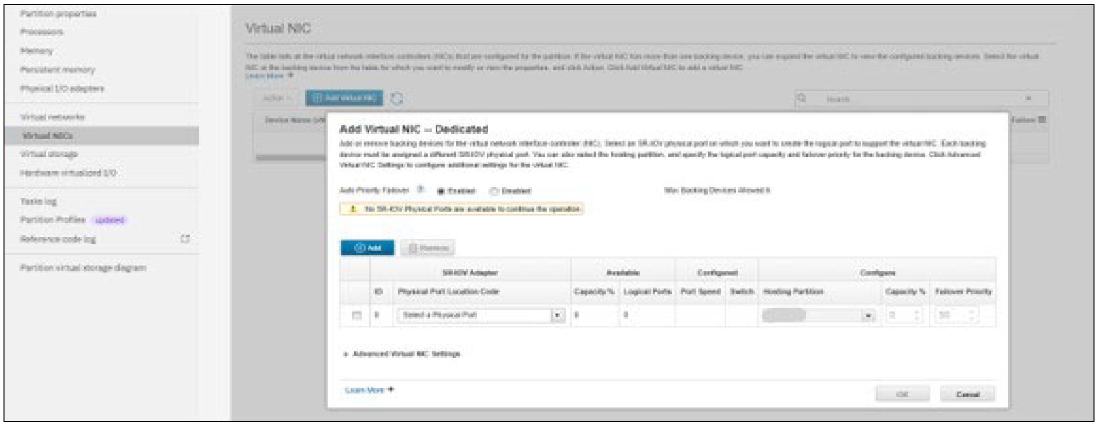


The table lists all the virtual network interface controllers (NICs) that are configured for the partition. If the virtual NIC has more than one backing device, you can expand the virtual NIC to view the configured backing devices. Select the virtual NIC or the backing device from the table for which you want to modify or view the properties, and click Action. Click Add Virtual NIC to add a virtual NIC.

Device Name (vNIC Adapter ID)	Auto Priority Failover	Capacity %	Backing Device Location Code	Port Label	Port Sub-label	Backing Device State	Hosting Partition	Failover

Figure 5-61 HMC Virtual NIC configuration

Add the virtual NIC on the client LPAR as shown in Figure 5-62 on page 170. When creating a virtual Network Interface Controller (vNIC) using Single Root I/O Virtualization (SR-IOV) on an IBM Power system, the Hardware Management Console (HMC) guides you through the necessary steps.



Add Virtual NIC -- Dedicated

Auto Priority Failover: ● Enabled ○ Disabled

Max Backing Devices Allowed 6

No SR-IOV Physical Ports are available to continue the operation

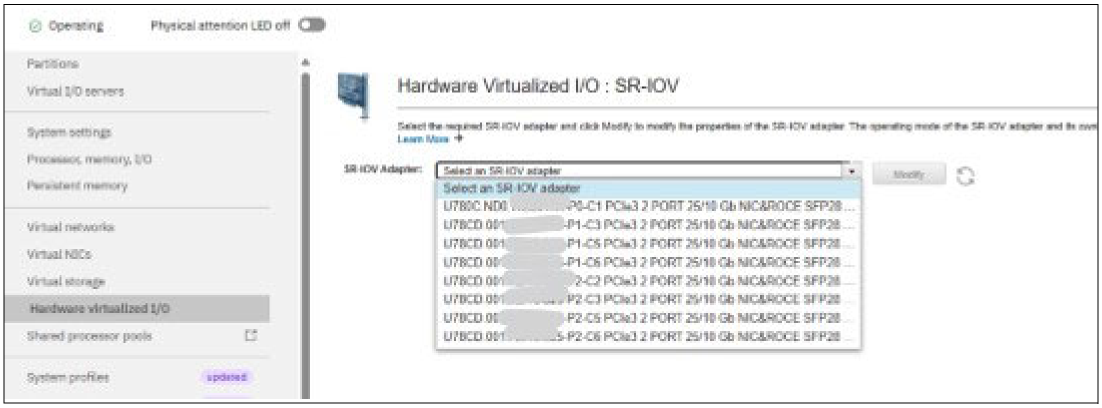
	SR-IOV Adapter		Available		Configured		Configure		
	ID	Physical Port Location Code	Capacity %	Logical Ports	Port Speed	Switch	Hosting Partition	Capacity %	Failover Priority
□	1	Select a Physical Port	0	0			[redacted]	0	50

Figure 5-62 Adding a virtual NIC

First, the HMC displays a list of Virtual I/O Servers (VIOS) that are eligible to host the SR-IOV port. This selection determines which VIOS will manage the vNIC's connection to the physical network. Next, you must choose a specific port on an available SR-IOV adapter.

The HMC performs a crucial check to ensure that a suitable SR-IOV adapter is present. If no adapter is found that supports SR-IOV, or if the available adapters are not configured correctly, the HMC will issue an alert. Specifically, an SR-IOV adapter must be placed into either “permissive” or “shared” mode to enable its use with vNICs. These modes allow the adapter to be virtualized and shared among multiple logical partitions, which is essential for SR-IOV functionality.

From the managed system in the HMC, modify an SR-IOV adapter to be in shared mode. This is shown in Figure 5-63.



Hardware Virtualized I/O : SR-IOV

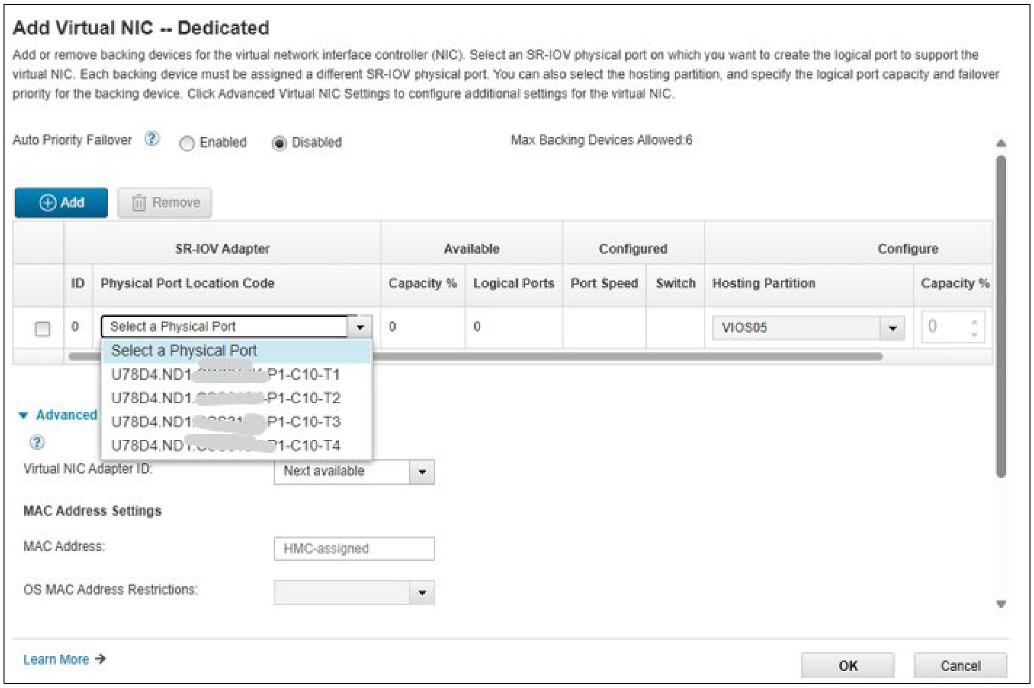
Select the required SR-IOV adapter and click Modify to modify the properties of the SR-IOV adapter. The operating mode of the SR-IOV adapter and its ove

SR-IOV Adapter: Select an SR-IOV adapter

Select an SR-IOV adapter
U78DC.ND0 [redacted] -P0-C1 PCIe3 2 PORT 25/10 Gb NIC&ROCE SFP28 ...
U78CD.001 [redacted] -P1-C3 PCIe3 2 PORT 25/10 Gb NIC&ROCE SFP28 ...
U78CD.001 [redacted] P1-C5 PCIe3 2 PORT 25/10 Gb NIC&ROCE SFP28 ...
U78CD.00 [redacted] -P1-C6 PCIe3 2 PORT 25/10 Gb NIC&ROCE SFP28 ...
U78CD.001 [redacted] 2-C2 PCIe3 2 PORT 25/10 Gb NIC&ROCE SFP28 ...
U78CD.001 [redacted] P2-C3 PCIe3 2 PORT 25/10 Gb NIC&ROCE SFP28 ...
U78CD.00 [redacted] P2-C5 PCIe3 2 PORT 25/10 Gb NIC&ROCE SFP28 ...
U78CD.001 [redacted] 5-P2-C6 PCIe3 2 PORT 25/10 Gb NIC&ROCE SFP28 ...

Figure 5-63 Setting adapter to shared mode

At this point, you can select a VIOS and an available SR-IOV adapter for the HMC to create a vNIC on as shown in Figure 5-64 on page 171.



Add Virtual NIC -- Dedicated

Add or remove backing devices for the virtual network interface controller (NIC). Select an SR-IOV physical port on which you want to create the logical port to support the virtual NIC. Each backing device must be assigned a different SR-IOV physical port. You can also select the hosting partition, and specify the logical port capacity and failover priority for the backing device. Click Advanced Virtual NIC Settings to configure additional settings for the virtual NIC.

Auto Priority Failover ○ Enabled ● Disabled   Max Backing Devices Allowed:6

	SR-IOV Adapter		Available		Configured		Configure	
	ID	Physical Port Location Code	Capacity %	Logical Ports	Port Speed	Switch	Hosting Partition	Capacity %
□	0	Select a Physical Port	0	0			VIOS05	0

Select a Physical Port
U78D4.ND1[redacted]P1-C10-T1
U78D4.ND1.[redacted]-P1-C10-T2
U78D4.ND1[redacted]P1-C10-T3
U78D4.ND[redacted]P1-C10-T4

Virtual NIC Adapter ID: Next available

MAC Address Settings

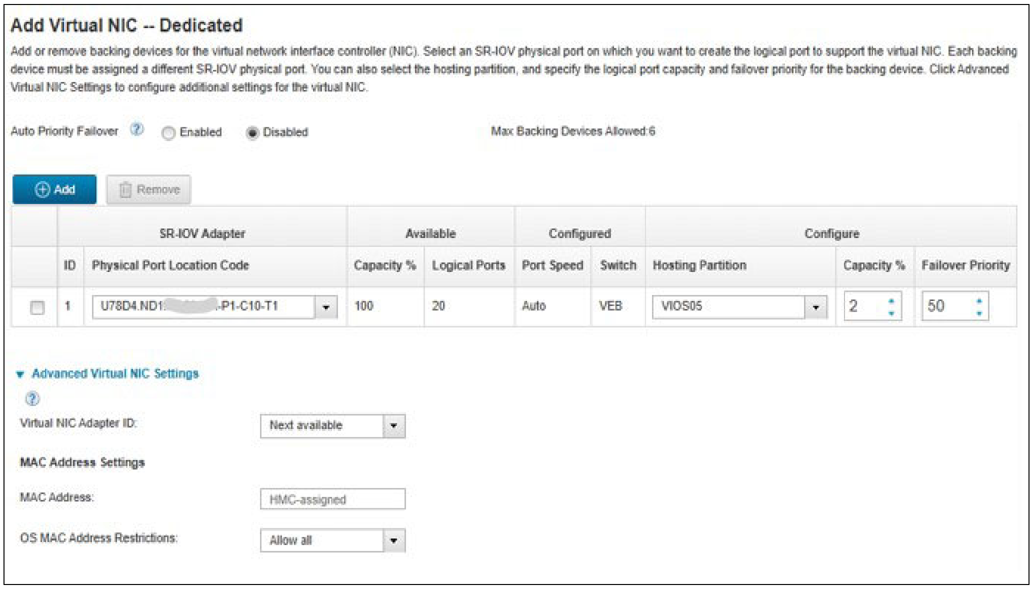
MAC Address: HMC-assigned

OS MAC Address Restrictions:

Figure 5-64 Choosing the VIOS and adapter

You can specify a percentage capacity allocation, enabling Quality of Service (QoS) management. This feature allows you to reserve a specific portion of the available bandwidth for the vNIC, ensuring consistent performance for critical applications. Furthermore, you can define a failover priority. This setting is crucial in vNIC failover configurations, where multiple vNICs are configured for redundancy. By assigning priorities, you designate which vNIC will act as the primary interface, handling network traffic under normal conditions, and which vNICs will serve as backups, automatically taking over in the event of a primary vNIC failure.

Figure 5-65 shows setting these values.



Add Virtual NIC -- Dedicated

Add or remove backing devices for the virtual network interface controller (NIC). Select an SR-IOV physical port on which you want to create the logical port to support the virtual NIC. Each backing device must be assigned a different SR-IOV physical port. You can also select the hosting partition, and specify the logical port capacity and failover priority for the backing device. Click Advanced Virtual NIC Settings to configure additional settings for the virtual NIC.

Auto Priority Failover ○ Enabled ● Disabled   Max Backing Devices Allowed:6

	SR-IOV Adapter		Available		Configured		Configure		
	ID	Physical Port Location Code	Capacity %	Logical Ports	Port Speed	Switch	Hosting Partition	Capacity %	Failover Priority
□	1	U78D4.ND1[redacted]-P1-C10-T1	100	20	Auto	VEB	VIOS05	2	50

Virtual NIC Adapter ID: Next available

MAC Address Settings

MAC Address: HMC-assigned

OS MAC Address Restrictions: Allow all

Figure 5-65 Defining details of vNIC

Select the required SRIOV adapter from the SR-IOV Adapter list to view the details of the selected adapter. Detailed information on configuring the vNIC can be found in this technote.

Viewing vNIC Information

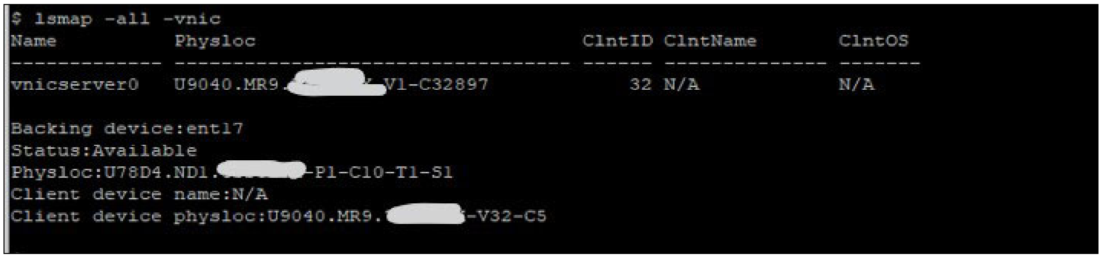
When an adapter is assigned to the VIOS as a vNIC capable adapter it will be displayed as a Converged Network Adapter VF adapter. However, when the adapter is allocated dynamically, this will only show after a Hypervisor update is performed by shutting down and restarting the VIOS. Example 5-43 shows the output from an **lsdev** command.

Example 5-43 Output from 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 (l-lan)
ent9        Available       Virtual I/O Ethernet Adapter (l-lan)
ent10       Available       Virtual I/O Ethernet Adapter (l-lan)
ent11       Available       Virtual I/O Ethernet Adapter (l-lan)
ent12       Available       Virtual I/O Ethernet Adapter (l-lan)
ent13       Available       Virtual I/O Ethernet Adapter (l-lan)
ent14       Available       Virtual I/O Ethernet Adapter (l-lan)
ent15       Available       Virtual I/O Ethernet Adapter (l-lan)
ent16       Available       Virtual I/O Ethernet Adapter (l-lan)
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 (l-lan)
ent22       Available       Virtual I/O Ethernet Adapter (l-lan)
ent23       Available       Virtual I/O Ethernet Adapter (l-lan)
```

When a Virtual I/O Server (VIOS) already utilizes a Shared Ethernet Adapter (SEA) for network virtualization, the behavior of SR-IOV capable adapters assigned for vNICs differs from traditional adapter visibility. Specifically, these SR-IOV adapters, dedicated to vNICs on Logical Partitions (LPARs), will not appear within the VIOS's profile under the standard I/O adapters list when an SEA is present. Instead, they remain exclusively within the SR-IOV Logical Ports menu. Importantly, this menu does not require any manual configuration, as the HMC automates the vNIC implementation directly from the client LPAR. This scenario assumes we're leveraging pure SR-IOV vNICs, without employing Hybrid Network Virtualization (HNV) or dedicating individual logical ports from the SR-IOV adapter.

To verify vNIC assignments, you can use the lsmap command on the VIOS incorporating the -*vnic* parameter as shown in Figure 5-66 on page 173. This command provides immediate visibility into the vNIC details, directly following the dynamic assignment of the SR-IOV vNIC capable adapter to both the VIOS and the LPAR. Crucially, this information is available without requiring a system reboot, offering real-time insight into the network configuration.



```
$ lsmap -all -vnic
Name          Physloc                            ClntID ClntName       ClntOS
------------- ---------------------------------- ------ -------------- -------
vnicserver0   U9040.MR9.[redacted]V1-C32897          32 N/A            N/A

Backing device:ent17
Status:Available
Physloc:U78D4.ND1.[redacted]-P1-C10-T1-S1
Client device name:N/A
Client device physloc:U9040.MR9.[redacted]-V32-C5
```

Figure 5-66 Using the lsmap command on the VIOS

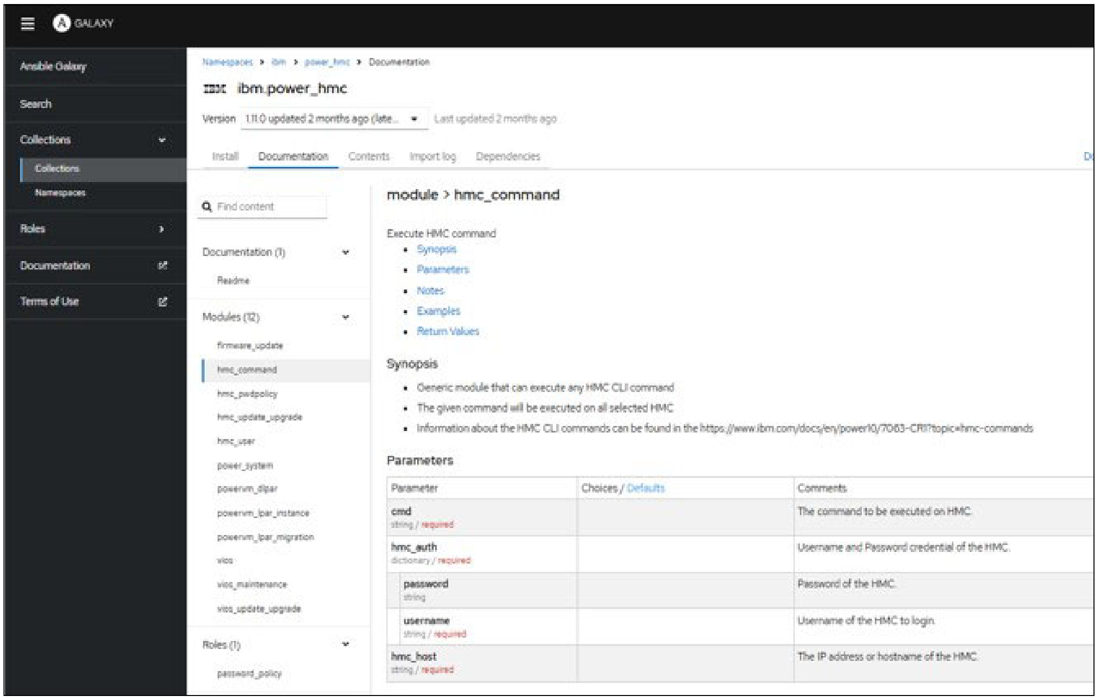
More detailed information is available in the HMC command line using **lshwres** as shown in Example 5-44. You could also opt to implement vNIC from the HMC command line and maintain a record of that implementation in a version controlled file.

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,curr_mode=ded,au
to_priority_failover=1,port_vlan_id=10,pvid_priority=0,allowed_vlan_ids=all,mac_addr=024ff663f70
5,allowed_os_mac_addrs=all,"backing_devices=sriov/VIOS06/2/4/0/27010002/2.0/2.0/50/100.0/100.0,s
riov/VIOS05/1/1/0/27004002/2.0/2.0/50/100.0/100.0","backing_device_states=sriov/27010002/0/Link
Down,sriov/27004002/1/Link Down"
```

vNIC and Automation

All management tasks for the vNIC are available in the HMC. However, automated tasks are available with the Ansible ibm.power_hmc collection using the cmd module and the HMC CLI. Figure 5-67 shows the IBM Power HMC collection in Ansible Galaxy.



ibm.power_hmc

Version 1.11.0 updated 2 months ago (late... Last updated 2 months ago

module > hmc_command

Execute HMC command

- Synopsis
- Parameters
- Notes
- Examples
- Return Values

Synopsis

- 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/power10/7063-CR1?topic=hmc-commands

Parameters

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 HNV and does not use a dedicated SR-IOV port to a client LPAR. The vNIC implementation virtualizes the Logical port using Virtual Functions. (VF).

Once an LPAR is using a client vNIC adapter, any target system for LPM tasks must have an adapter in SR-IOV shared mode with an available logical port and available capacity on a physical port.

> **Note:** Hybrid Network Virtualization leverages existing technologies such as AIX Network Interface Backup (NIB) and IBM i Virtual IP Address (VIPA) as its foundation

vNIC Failover

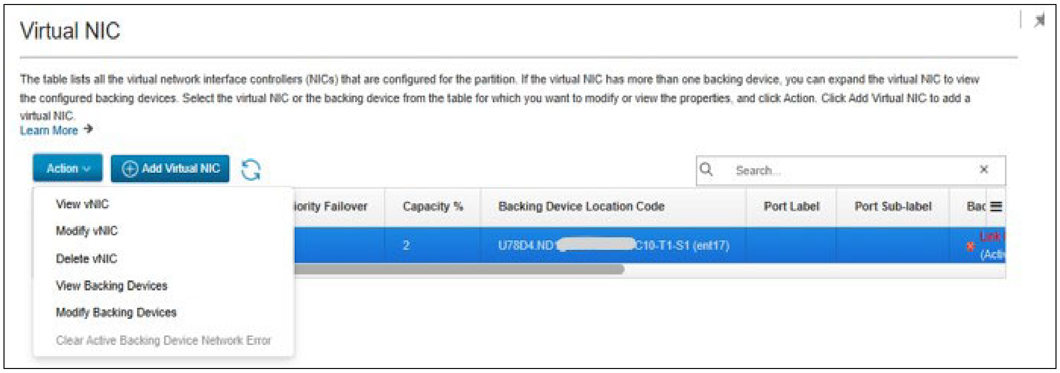
VNIC Failover is a vNIC with multiple backing devices for redundancy (analogous to SEA Failover). vNIC Failover allows a vNIC client to be configured with up to 6 backing devices. One backing device is active while the others are inactive standby devices. If the Power Hypervisor detects the active backing device is no longer operational a failover is initiated to the most favored (lowest Failover Priority value) operational backing device.

Power10 Servers In general, HMC, system firmware, and operating systems with support for Power10 Systems include support for vNIC Failover.

A vNIC backing device consists of a SR-IOV shared mode enabled adapter physical port, a SR-IOV logical port, a vNIC server virtual adapter, and a Virtual I/O Server (VIOS).

During configuration of a vNIC backing device, the user selects a VIOS and adapter physical port. When the backing device is instantiated the management console creates a vNIC server and a logical port for the physical port on the VIOS. Once created, the backing device is available as either an active or standby backing device for its vNIC client. The logical port and vNIC server adapter are associated with a single vNIC client.

In the HMC for the Virtual NIC attached to the LPAR, you can modify the configuration as shown in Figure 5-68.



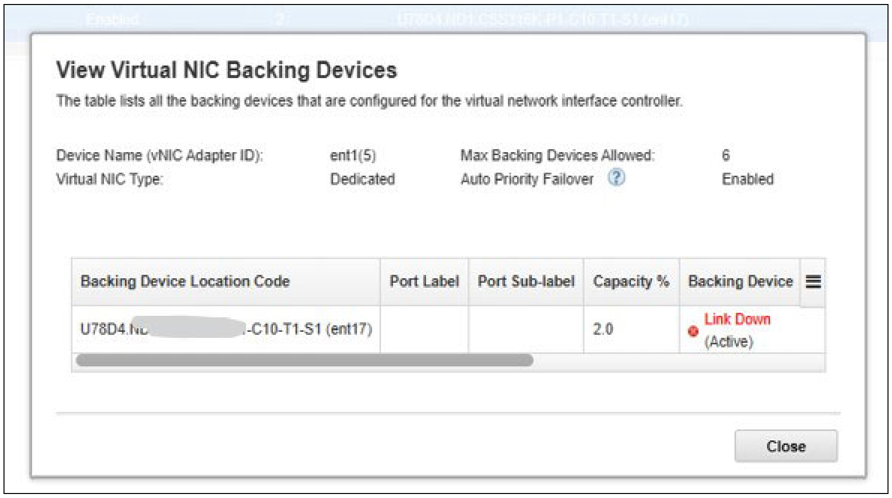
Virtual NIC

The table lists all the virtual network interface controllers (NICs) that are configured for the partition. If the virtual NIC has more than one backing device, you can expand the virtual NIC to view the configured backing devices. Select the virtual NIC or the backing device from the table for which you want to modify or view the properties, and click Action. Click Add Virtual NIC to add a virtual NIC.

iority Failover	Capacity %	Backing Device Location Code	Port Label	Port Sub-label	Bac
	2	U78D4.ND[redacted]C10-T1-S1 (ent17)			Link (Acti

Figure 5-68 Virtual NIC configuration

When you view the vNIC backing devices, you will see the vNICs assigned to this LPAR as shown in Figure 5-69.



View Virtual NIC Backing Devices

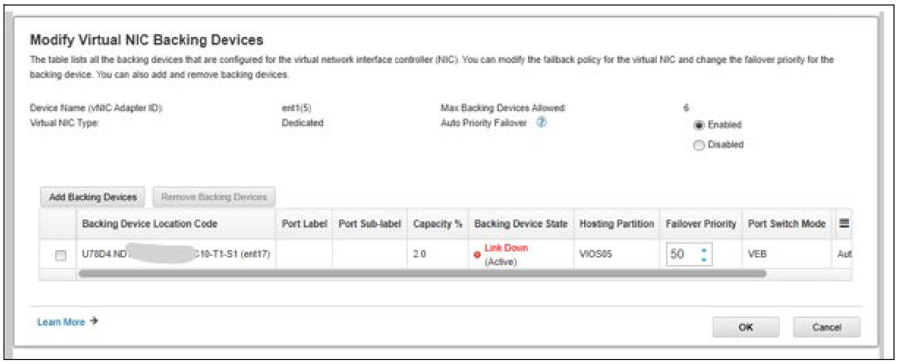
The table lists all the backing devices that are configured for the virtual network interface controller.

Device Name (vNIC Adapter ID):	ent1(5)	Max Backing Devices Allowed:	6
Virtual NIC Type:	Dedicated	Auto Priority Failover	Enabled

Backing Device Location Code	Port Label	Port Sub-label	Capacity %	Backing Device
U78D4.ND [redacted] -C10-T1-S1 (ent17)			2.0	Link Down (Active)

Figure 5-69 Viewing available backing devices

In the Modify backing device option, you can add a backing device to the existing vNIC as shown in Figure 5-70.



Modify Virtual NIC Backing Devices

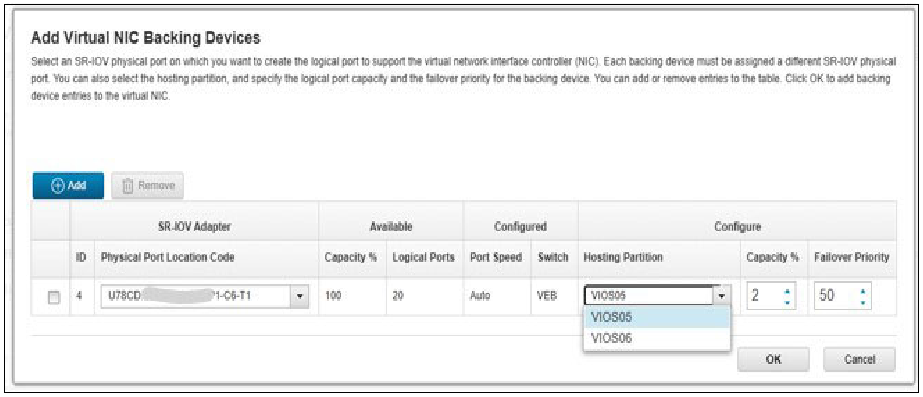
The table lists all the backing devices that are configured for the virtual network interface controller (NIC). You can modify the failback policy for the virtual NIC and change the failover priority for the backing device. You can also add and remove backing devices.

Device Name (vNIC Adapter ID):	ent1(5)	Max Backing Devices Allowed:	6
Virtual NIC Type:	Dedicated	Auto Priority Failover	● Enabled ○ Disabled

	Backing Device Location Code	Port Label	Port Sub-label	Capacity %	Backing Device State	Hosting Partition	Failover Priority	Port Switch Mode	
□	U78D4.ND [redacted] C10-T1-S1 (ent17)			2.0	Link Down (Active)	VIOS05	50	VEB	Aut

Figure 5-70 Adding backing device to vNIC

With more than one SR-IOV capable adapters configured in sharing mode for the managed system you will be able to select the other VIOS as a hosting partition for the other vNIC backing device as shown in Figure 5-71.



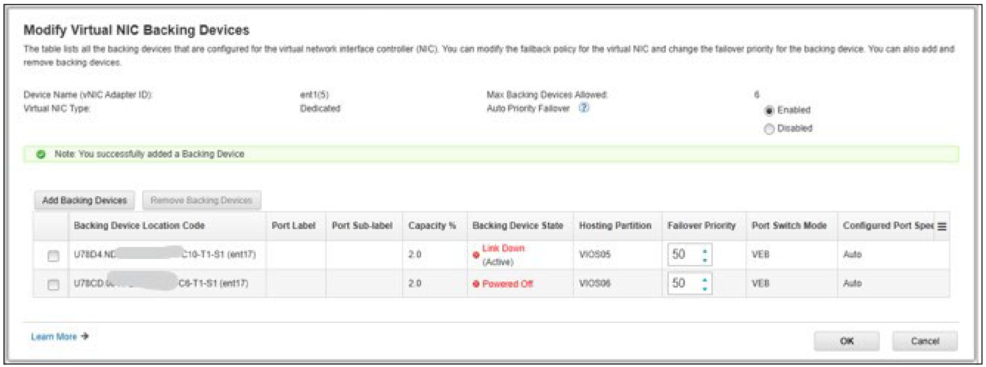
Add Virtual NIC Backing Devices

Select an SR-IOV physical port on which you want to create the logical port to support the virtual network interface controller (NIC). Each backing device must be assigned a different SR-IOV physical port. You can also select the hosting partition, and specify the logical port capacity and the failover priority for the backing device. You can add or remove entries to the table. Click OK to add backing device entries to the virtual NIC.

	SR-IOV Adapter		Available		Configured		Configure		
	ID	Physical Port Location Code	Capacity %	Logical Ports	Port Speed	Switch	Hosting Partition	Capacity %	Failover Priority
□	4	U78CD [redacted] 1-C6-T1	100	20	Auto	VEB	VIOS05	2	50

Figure 5-71 Selecting VIOS for failover

The Hypervisor will manage failover by using the failover priority attribute of the vNIC. When set to automatic, will enable the Hypervisor to decide which VIOS has priority for network traffic and which VIOS to switch to for availability as shown in Figure 5-72.



Modify Virtual NIC Backing Devices

The table lists all the backing devices that are configured for the virtual network interface controller (NIC). You can modify the failback policy for the virtual NIC and change the failover priority for the backing device. You can also add and remove backing devices.

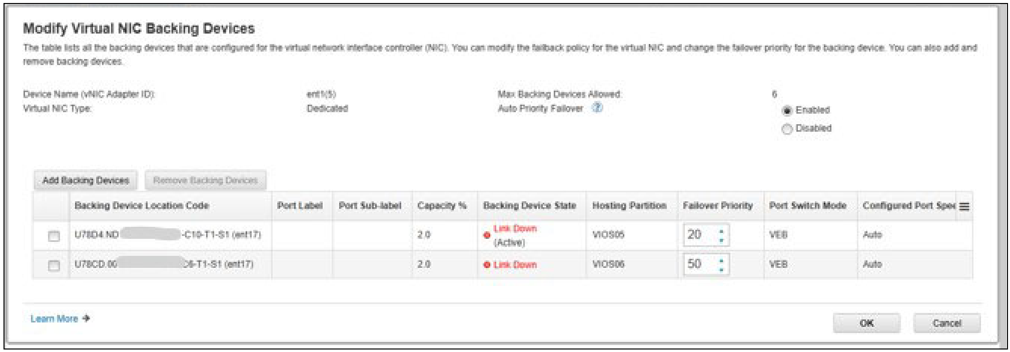
Device Name (vNIC Adapter ID): ent1(5)
Virtual NIC Type: Dedicated
Max Backing Devices Allowed: 6
Auto Priority Failover: ● Enabled ○ Disabled

Note: You successfully added a Backing Device

	Backing Device Location Code	Port Label	Port Sub-label	Capacity %	Backing Device State	Hosting Partition	Failover Priority	Port Switch Mode	Configured Port Spec
□	U78D4.ND[redacted]C10-T1-S1 (ent17)			2.0	Link Down (Active)	VIOS05	50	VEB	Auto
□	U78CD.[redacted]C6-T1-S1 (ent17)			2.0	Powered Off	VIOS06	50	VEB	Auto

Figure 5-72 Setting failover priority

You can modify the priority of each vNIC dynamically after configuration by going into the Modify Virtual NIC backing Devices again as shown in Figure 5-73.



Modify Virtual NIC Backing Devices

The table lists all the backing devices that are configured for the virtual network interface controller (NIC). You can modify the failback policy for the virtual NIC and change the failover priority for the backing device. You can also add and remove backing devices.

Device Name (vNIC Adapter ID): ent1(5)
Virtual NIC Type: Dedicated
Max Backing Devices Allowed: 6
Auto Priority Failover: ● Enabled ○ Disabled

	Backing Device Location Code	Port Label	Port Sub-label	Capacity %	Backing Device State	Hosting Partition	Failover Priority	Port Switch Mode	Configured Port Spec
□	U78D4.ND[redacted]-C10-T1-S1 (ent17)			2.0	Link Down (Active)	VIOS05	20	VEB	Auto
□	U78CD.00[redacted]C6-T1-S1 (ent17)			2.0	Link Down	VIOS06	50	VEB	Auto

Figure 5-73 Modify vNIC priority

Information on the vNIC baking devices is also viewable using the **`entstat`** command. You will only see one VIOS detailed in “Server information”. The VIOS that is currently providing the SR-IOV path. This is shown in Example 5-45.

Example 5-45 Displaying 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 will be 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
hdcrypt Available       Data encryption
```

```
pkcs11  Available       ACF/PKCS#11 Device
vsa0    Available       LPAR Virtual Serial Adapter
vscsi0  Available       Virtual SCSI Client Adapter
root@workstation3:/>
```

vNIC VIOS Resource Requirements

Table 5-4 provides estimates for memory usage by the Power Hypervisor and VIOS for vNIC and vNIC Failover. Actual memory usage may vary by configuration.

Table 5-4 Estimated resource allocation for vNIC

Adapter FC	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, EN18	160 MB	9 MB	0.7 MB	7.5 MB
EC3L, EC3M,	3.7 GB	9 MB	0.7 MB	25 MB
EC2T, EC2U, EC2R, EC2S	2.9 GB	9 MB	0.7 MB	25 MB
EC66, EC67	5.4 GB	9 MB	0.7 MB	25 MB
EC75, EC76	5.9 GB	9 MB	0.7 MB	25 MB
EC71, EC72	2.9 GB	9 MB	0.7 MB	25 MB
EN24, EN26	4.5 GB	9 MB	0.7 MB	25 MB

It is important to understand that CPU utilization on a Virtual I/O Server (VIOS) handling vNIC traffic is highly variable, directly influenced by the nature of the network load. While a general rule of thumb suggests allocating approximately 0.7 additional VIOS cores per 10Gb/s of bandwidth for peak bandwidth scenarios involving large packets, this should be considered a baseline. Workloads characterized by high message rates and smaller packet sizes will inevitably demand significantly greater CPU resources. This is due to the increased processing overhead associated with handling a larger volume of individual network packets. Therefore, careful monitoring and capacity planning are essential to ensure the VIOS can effectively manage the network traffic without becoming a bottleneck.

6

How to modernize your applications

Modernizing applications on IBM Power is about transforming existing assets to thrive in a rapidly changing technological environment. As you work to modernize your applications running on IBM Power focus on:

- Leveraging the strengths of your existing applications. IBM Power systems are known for their reliability, performance, and security. Modernization efforts aim to preserve these strengths while adding new capabilities.
- Embracing hybrid cloud capabilities. Modernization often involves integrating on-premises Power systems with cloud-based resources, creating a hybrid cloud environment.
- Utilizing containerization and Kubernetes. Technologies like Red Hat OpenShift enable containerization, allowing applications to be broken down into smaller, more manageable components. This facilitates agility and scalability.
- Focus on cloud-native development for new functions. Modernization encourages the adoption of cloud-native development practices, including microservices, APIs, and DevOps.
- Integrating AI. IBM Power is well-suited for AI workloads. and you should focus on integrating AI capabilities into existing applications.
- Focus on data, IBM Power is known as being an excellent platform for data access, providing performance and reliability for your data. Modern applications need to be able to handle and analyze vast amounts of data.

The following topics are discussed in this chapter:

© Copyright IBM Corp. 2025.

6.1 Modernizing legacy systems through Domain Segregation

This section explores the challenges of modernizing legacy systems and presents domain segregation as a powerful strategy for overcoming these obstacles. We will discuss the principles of domain-driven design (DDD), its application in decomposing monolithic systems, and the benefits of this approach for improving maintainability, scalability, and adaptability.

Modernizing legacy systems is a critical challenge for many organizations. These systems, often built with monolithic architectures, struggle to keep pace with the rapidly evolving demands of today's digital world. They are often characterized by tight coupling, limited scalability, slow development cycles, and difficulty in adapting to change.

Domain segregation, inspired by DDD principles, offers a powerful approach to address these challenges. It involves dividing the system into smaller, self-contained units, each with its own distinct business logic and data, establishing clear boundaries between these contexts, and implementing each bounded context as a separate microservice, enabling independent development, deployment, and scaling.

Domain segregation significantly improves maintainability by breaking down the system into smaller, more 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 utilization. The ability to independently deploy and update microservices enables faster delivery of new features and quicker responses to changing business needs. By isolating failures within individual domains, domain segregation helps to improve system resilience and reduce the impact of outages.

Implementing domain segregation involves identifying bounded contexts by analyzing the current system to identify 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, and continuously monitoring and refining the domain boundaries as needed.

Challenges and considerations include maintaining data consistency across multiple microservices, managing the increased complexity of a distributed system with multiple microservices, and the need for careful planning and execution for thorough testing and debugging of interactions between microservices.

Domain segregation offers a promising approach for modernizing legacy systems. By breaking down monolithic systems into smaller, more manageable units, organizations can improve their agility, scalability, and resilience while reducing development and maintenance costs. While challenges exist, the benefits of this approach often outweigh the risks, enabling businesses to adapt more effectively to the ever-changing demands of the digital age.

Domain Model Definition (DMD): A Blueprint for Microservices

The Domain Model Definition (DMD) is a pragmatic approach to structuring complex software systems, particularly those adopting a microservices architecture. It acts as a blueprint, guiding the decomposition of a system into smaller, more manageable domains.

Traditional data models often suffer from complex interconnections, hindering maintainability and adaptability. Designing and implementing independent microservices requires a clear understanding of domain boundaries and interactions.

The DMD promotes the division of the system into distinct, self-contained domains, each with its own set of data and logic. The DMD is defined using a flexible format (e.g., YAML, XML, JSON), making it independent of specific programming languages, databases, or development tools. However, mature design tools typically use UML. The DMD is not a static artifact. It evolves iteratively as the system grows and changes, accommodating new requirements and addressing unforeseen challenges.

By breaking down the system into smaller, more focused domains, the DMD enhances maintainability and reduces the risk of unintended side effects. The platform-agnostic nature of the DMD allows for greater flexibility in technology choices and easier adaptation to evolving business needs. The DMD provides a clear foundation for designing and implementing microservices, ensuring that they are well-defined, independent, and aligned with business requirements.

The implementation process involves gathering relevant information about the system, including data models, existing applications, and business requirements. Based on the gathered information, distinct domains within the system are identified and defined, considering factors such as business logic, data dependencies, and team responsibilities. The defined domains are then translated into concrete implementation artifacts, such as microservices, data models, and APIs. Continuous monitoring and evaluation of the system are crucial, with the DMD being adapted as necessary to accommodate changes in requirements, technology, or organizational structure.

The DMD is a valuable tool for organizations embracing a microservices architecture. By providing a structured and flexible approach to domain modeling, it empowers development teams to create robust, scalable, and maintainable systems that effectively meet the evolving needs of the business.

Domain Model Definition (DMD): A Deeper Dive

The Domain Model Definition (DMD) is more than just a simple data structure; it's a strategic blueprint for building complex software systems, particularly those embracing the microservices architecture.

The DMD is deeply rooted in DDD principles. It emphasizes understanding the core business domain, identifying key concepts and their relationships, and translating this knowledge into a software model. The DMD helps define “bounded contexts,” which are self-contained units within the system with their own unique language and rules. This promotes modularity and reduces complexity. The DMD encourages the use of a shared language between business stakeholders and developers. This common vocabulary ensures clear communication and avoids misunderstandings.

A conceptual representation of the core business domain, capturing key entities, their attributes, and relationships. Explicitly defined boundaries that separate different domains within the system. These boundaries help to isolate changes and reduce the impact of modifications. A representation of how data is stored and accessed within each domain. This may include database schema, data structures, and APIs. Specifications for the services that interact with each domain. These definitions include service interfaces, data formats, and communication protocols.

The DMD fosters better communication and collaboration between business stakeholders and development teams. By breaking down the system into smaller, more manageable domains, the DMD improves maintainability and reduces the risk of unintended side effects. The modular structure enabled by the DMD allows for faster and more flexible development and deployment of new features. The clear separation of concerns facilitated by the DMD makes it easier to test individual components and the system as a whole.

The DMD can be represented using various formats, including UML diagrams, textual descriptions, or specialized modeling tools. The DMD is an evolving artifact. It should be continuously refined as the system evolves and new requirements emerge. Utilizing tools and automation can streamline the process of generating code, deploying services, and managing the DMD itself.

Example

Imagine an e-commerce system. The DMD might define separate domains for:

- Product Catalog

  Managing product information, pricing, and inventory.

- Order Management

  Handling order processing, fulfillment, and shipping.

- Customer Relationship Management (CRM)

  Managing customer accounts, preferences, and interactions.

Each domain would have its own data model, services, and potentially even its own development team, enabling independent development and deployment.

Modernization Strategy: Gradual Microservices with Shared Database

A phased approach to microservices modernization can mitigate the risk of overwhelming the project.

Begin by developing new microservices that interact with the existing monolithic database. This minimizes the initial impact on the legacy system and allows for a gradual transition. Utilize frameworks like Spring Boot, well-suited for both microservices and integration with existing databases.

Enforce strict adherence to domain boundaries. Microservices should never directly access data within another service's domain. All inter-service communication must occur through well-defined APIs. This ensures loose coupling, improves maintainability, and allows for independent scaling and deployment of services.

This phased approach offers several advantages:

- Reduced Risk: Minimizes disruption to the existing system and allows for incremental improvements.
- Faster Time-to-Market: Enables the delivery of new features and functionalities more quickly.
- Improved Agility: Provides greater flexibility to adapt to changing business requirements.

This strategy serves as a starting point. As the modernization progresses, consider strategies for database refactoring, such as data partitioning or creating dedicated databases for specific microservices, to further enhance performance and scalability.

Conclusion

The Domain Model Definition is a powerful tool for building complex software systems, particularly those adopting a microservices architecture. By providing a structured and well-defined framework for understanding and modeling the business domain, the DMD enables teams to create systems that are more maintainable, adaptable, and aligned with business needs.

6.2 Programming languages

Choosing the right programming language for your applications depends on several factors, including the nature of the project, performance requirements, scalability, developer expertise, and the ecosystem of libraries and tools available. Here are some key considerations:

- Project Requirements

  Different projects have different needs. For example, Python is excellent for data science, machine learning, and rapid prototyping due to its simplicity and extensive libraries like TensorFlow and Pandas. JavaScript is ideal for web development, especially with frameworks like React and Angular for front-end and Node.js for back-end development.

- Performance and Scalability

  If your application requires high performance and scalability, languages like Go and Rust are strong contenders. Go is known for its efficient concurrency model, making it suitable for cloud services and backend systems. Rust offers memory safety and performance, making it ideal for system-level programming and applications where control over hardware resources is crucial.

- Developer Expertise

  The skill set of your development team plays a significant role. If your team is experienced in a particular language, it might be more efficient to use that language. For instance, if your team is well-versed in RPG, leveraging modern RPG on IBM i can be beneficial for maintaining and modernizing legacy systems.

- Ecosystem and Libraries

  The availability of libraries and frameworks can greatly influence your choice. Python's vast ecosystem supports a wide range of applications, from web development to scientific computing. JavaScript's integration with numerous web technologies makes it indispensable for web developers. Similarly, Go and Rust have growing ecosystems that support modern development needs.

- Integration with Existing Systems

  Consider how well the language integrates with your existing infrastructure. For example, modern RPG on IBM i can seamlessly interface with other languages and tools, making it a good choice for modernizing legacy systems while maintaining reliability.

- 6. Community and Support

  A strong community can provide valuable support and resources. Languages like Python and JavaScript have large, active communities that contribute to a wealth of tutorials, documentation, and third-party libraries.

Ultimately, the right programming language for your application will depend on balancing these factors to meet your specific needs. It's often beneficial to use a combination of languages to leverage their respective strengths for different parts of your application.

6.2.1 Integrating traditional and modern technologies

Combining traditional programming languages like RPG and COBOL with modern development approaches on IBM Power Systems can be a strategic way to leverage existing investments while embracing newer technologies. One effective method is to extend legacy applications using microservices and modern languages – enhancing capabilities without rewriting everything from scratch.

However, this hybrid modernization approach comes with ongoing challenges, particularly in maintaining and updating existing RPG and COBOL codebases. Key issues include:

- Shrinking Talent Pool

  As veteran RPG and COBOL developers retire, the pool of experienced professionals continues to decline. This expertise gap makes it increasingly difficult to maintain and evolve legacy systems.

- Attracting New Talent

  Modern developers tend to favor languages like Python, JavaScript, and Go, which offer more contemporary features and wider applicability. This preference makes it harder to recruit fresh talent for IBM i environments.

- Training and Education

  Bridging the skills gap requires significant investment in training programs—for both upskilling current staff and educating new hires on traditional languages and IBM i platform specifics. Though time-consuming and costly, this is crucial for successful modernization.

- Integration with Modern Technologies

  Modernizing IBM Power Systems often involves integrating legacy applications with cloud platforms, AI-driven analytics, and web interfaces. Developers need proficiency in both traditional languages and modern tools to ensure seamless operation and interoperability.

Tools to Aid Transition

Tools like IBM Rational Developer for i (RDi) and IBM's Modernization Engine for Lifecycle Integration (MERLIN) help convert fixed-format RPG code to modern free-form RPG, making legacy code more accessible and manageable.

The Role of Generative AI

Emerging AI tools like IBM's RPG Code Assistant are transforming modernization workflows. These tools help developers interpret legacy code, generate new functionality from natural language prompts, and even create test cases automatically—significantly accelerating development and reducing manual effort.

Conclusion

Addressing these challenges is essential for organizations aiming to modernize their IBM Power Systems without compromising the performance and reliability of their legacy applications. Leveraging both traditional strengths and modern innovation offers a balanced, future-ready path forward.

6.2.2 Choosing a modern language for your project

Python is one of the most popular modern programming languages, known for its simplicity, readability, and vast ecosystem of libraries and frameworks. Its clean syntax allows developers to focus on solving problems rather than dealing with complex language rules. Python is often used in fields like data science, artificial intelligence (AI), web development, and automation. Libraries like TensorFlow, Django, and Pandas make Python an excellent choice for scientific computing, machine learning, and web application development. Its versatility, along with its large community, makes Python a go-to language for rapid prototyping and production-grade solutions alike.

JavaScript, a language that initially gained traction for client-side web development, is now indispensable for building full-stack applications thanks to the rise of Node.js. JavaScript allows developers to create dynamic, responsive user interfaces and is the backbone of modern web development. With frameworks like React, Angular, and Vue.js, JavaScript powers front-end development, while Node.js brings it to the server side for building scalable network applications. One of its main advantages is its asynchronous, non-blocking nature, making it well-suited for high-performance, real-time applications such as chat apps or live updates. Furthermore, JavaScript's integration with numerous web technologies and tools makes it an essential skill for any web developer.

Go (or Golang) is a statically typed, compiled language developed by Google, known for its simplicity, performance, and efficiency, particularly in handling concurrency. Go is used extensively in cloud infrastructure, backend services, and microservices architecture. Its concurrency model, based on goroutines and channels, allows developers to write highly scalable and concurrent applications with ease. Go is particularly advantageous for building cloud-native applications, APIs, and systems that require low latency and high performance, making it a top choice for companies dealing with large-scale distributed systems. Its fast compilation times and strong focus on simplicity make it a favorite in DevOps and infrastructure automation projects.

Rust is a systems programming language designed for safety, performance, and concurrency. It provides memory safety without the need for garbage collection, making it ideal for performance-critical applications such as game engines, embedded systems, and operating systems. Rust's ownership model prevents common bugs like null pointer dereferencing and data races, which are typical in low-level languages like C and C++. Its focus on preventing memory-related errors during compile time ensures robust and safe software, making it a top choice for developers who need control over system resources while maintaining high levels of safety and concurrency.

Modern RPG on IBM i has evolved significantly from its origins as a column-based language to a versatile, free-form language that meets contemporary business needs. This transformation has been driven by the need to keep up with rapid changes in application development and to ensure that IBM i remains a robust platform for creating and developing applications. Modern RPG now supports IBM Integrated Language Environment® (ILE), allowing it to interface seamlessly with other languages and tools. Additionally, tools like IBM Rational® Developer for i (RDi) and the IBM i Modernization Engine (MERLIN) automate the conversion of fixed-format RPG code to modern free-form RPG, making it easier for new developers to understand and work with. This modernization effort ensures that IBM i continues to be a relevant and powerful system for businesses today.

In summary, modern programming languages like Python, JavaScript, Go, Rust, and modern RPG each offer distinct advantages tailored to different domains of software development. Python excels in ease of use and scientific computing, JavaScript dominates the web development world, Go shines in scalable backends and cloud services, Rust is preferred for system-level programming requiring high performance and safety, and modern RPG ensures the continued relevance and robustness of IBM i systems. Choosing the right modern language depends on the specific requirements of the project, such as performance, scalability, or ease of development. Each of these languages reflects the trends and demands of the modern software landscape, making them invaluable tools for developers across various industries.

6.2.3 Integrated Development Environments

Using an Integrated Development Environment (IDE) offers several advantages that can significantly enhance the productivity and efficiency of developers. Here are some key reasons to use an IDE:

- Code Editing and Navigation: IDEs provide advanced code editing features such as syntax highlighting, code completion, and error detection. These features help developers write code faster and with fewer errors. Additionally, IDEs offer powerful navigation tools that make it easy to jump between different parts of the codebase.
- Debugging: IDEs come with integrated debugging tools that allow developers to set breakpoints, inspect variables, and step through code execution. This makes it easier to identify and fix bugs, improving the overall quality of the software.
- Version Control Integration: Many IDEs integrate with version control systems like Git, allowing developers to manage their code repositories directly within the IDE. This integration simplifies tasks such as committing changes, branching, and merging code.
- Build and Deployment: IDEs often include tools for building and deploying applications. These tools automate the compilation process, package the application, and deploy it to the target environment, saving time and reducing the risk of errors.
- Project Management: IDEs provide project management features that help organize code files, libraries, and resources. This organization makes it easier to manage large projects and collaborate with other developers.
- Extensibility: Many IDEs support plugins and extensions that add additional functionality. Developers can customize their IDE to suit their specific needs, whether it's integrating new tools, adding support for different languages, or enhancing productivity features.
- Consistency: Using an IDE ensures a consistent development environment across different projects and team members. This consistency helps maintain coding standards and reduces the “works on my machine” problem.
- Productivity: Overall, IDEs streamline the development process by providing a comprehensive set of tools in one place. This reduces the need to switch between different applications and helps developers stay focused on their tasks.

In summary, IDEs are invaluable for improving code quality, enhancing productivity, and simplifying the development process. They provide a cohesive environment that supports all aspects of software development, from writing and debugging code to managing projects and deploying applications. Development Environments (IDEs) play a crucial role in developing and modernizing applications on IBM Power systems, including IBM i, AIX, and Linux, as well as platforms like OpenShift.

IBM i

IBM Rational Developer for i (RDi) is the primary IDE for IBM i development. Built on the Eclipse platform, RDi provides a comprehensive set of tools for creating, maintaining, and modernizing applications on IBM i. It supports various programming languages, including RPG, COBOL, Java, and C/C++. RDi offers features like language-aware source editing, visual analysis tools, and integrated compile error feedback, which streamline the development process. Additionally, RDi integrates with IBM Rational Team Concert® for better application lifecycle management.

AIX

For AIX, developers often use IDEs like IBM Open XL C/C++ and SlickEdit. IBM Open XL C/C++ is a next-generation compiler that facilitates the creation and maintenance of applications written in C/C++ for IBM Power platforms. It incorporates the LLVM and Clang compiler infrastructure, maximizing hardware utilization and performance 2. SlickEdit is another popular choice, offering robust editing capabilities and support for various programming languages, including C/C++, Java, and Python.

Linux

Linux offers a wide range of IDEs suitable for various development needs. Visual Studio Code (VS Code) is highly popular due to its flexibility, extensive extensions, and strong Git integration. It supports multiple programming languages and provides features like IntelliSense for code completion and debugging. IntelliJ IDEA is another powerful IDE, particularly favored for Java development. It offers intelligent code analysis, smart code completion, and seamless integration with various version control systems.

OpenShift

Red Hat OpenShift Dev Spaces is a cloud-based development environment built on the Eclipse Che project. It uses Kubernetes and containers to provide a consistent, secure, and zero-configuration development environment. OpenShift Dev Spaces supports IDEs like VS Code and JetBrains IntelliJ IDEA, allowing developers to code, build, test, and run applications directly in the browser. This setup eliminates the “works on my machine” problem by defining development environments as code, ensuring consistency and reproducibility. Additionally, Red Hat offers IDE extensions for OpenShift, enhancing the development experience by integrating OpenShift functionalities directly into VS Code and IntelliJ.

Summary

Choosing the right IDE for IBM Power systems depends on the specific requirements of the development environment. RDi is ideal for IBM i, offering specialized tools for RPG and COBOL development. For AIX, IBM Open XL C/C++ and SlickEdit provide robust options for C/C++ development. Linux developers benefit from versatile IDEs like VS Code and IntelliJ IDEA, while OpenShift Dev Spaces offers a modern, cloud-based development environment that integrates seamlessly with Kubernetes.

6.3 Database Technologies

IBM Power Systems are renowned for their performance, reliability, and scalability – traits that make them ideal platforms for database workloads. While many organizations have historically relied on IBM’s Db2, the platform has evolved to support a broad ecosystem of modern, open-source, and NoSQL databases.

6.3.1 Db2 – The Enterprise-Grade Relational Backbone

Db2 is a powerful relational database management system that is deeply optimized for Power architectures, particularly on AIX and IBM i. It excels in online transaction processing (OLTP), complex analytics, and enterprise integration. With features like high availability, disaster recovery, and mature tooling, Db2 remains a preferred choice for many enterprises. However, it adheres to a traditional relational model, which can be limiting when dealing with unstructured data or modern application demands that require more flexibility.

IBM Db2 itself has evolved to incorporate modern features, including JSON document storage, RESTful APIs, and integration with big data frameworks like Spark and Hadoop. This means organizations can use Db2 as both a traditional relational database and a NoSQL-style engine, allowing for modern flexibility without abandoning the IBM ecosystem. These capabilities make it possible to implement hybrid architectures that combine the strengths of relational and non-relational data models.

Db2 supports both transactional and analytical workloads, providing features like AI-powered query optimization, continuous availability, and robust security. It can be deployed on-premises, in the cloud, or in hybrid environments, offering flexibility to meet diverse business needs

6.3.2 Modern databases

Modern databases represent a significant evolution in data management, driven by the need to handle the volume, velocity, and variety of data generated in the digital age. Unlike traditional relational databases, which rely on structured tables and rigid schema, modern databases offer greater flexibility and scalability. They encompass a range of database technologies, including NoSQL, NewSQL, and cloud-native databases, each designed to address specific data management challenges. This diversity empowers organizations to choose the database solution that best fits their unique application requirements.

One of the key advantages of modern databases is their ability to scale horizontally. This means that 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, like MongoDB and Cassandra, are well-suited for applications that require high throughput and low latency, such as real-time analytics, IoT applications, and mobile backends. NewSQL databases, such as CockroachDB and YugabyteDB, combine the scalability of NoSQL with the ACID properties of traditional relational databases, making them ideal for financial applications and e-commerce platforms. Cloud-native databases, such as Amazon DynamoDB, Azure Cosmos DB, and Google Cloud Spanner, offer fully managed, highly available, and globally distributed database services, simplifying database administration and ensuring business continuity.

In conclusion, modern databases provide a powerful toolkit for managing data in today's complex and dynamic environment. Their scalability, flexibility, and diverse capabilities enable organizations to build innovative applications, gain valuable insights, and drive business growth. By understanding the strengths and weaknesses of different database technologies, businesses can make informed decisions and leverage the right tools to unlock the full potential of their data.

6.3.3 Summary and Comparison

IBM Power Systems are no longer just homes for legacy relational data—they're open platforms that support a broad range of database technologies, from traditional OLTP systems to real-time NoSQL stores and containerized cloud-native databases.

Whether you're running transactional systems on Db2 for i, real-time web services with MongoDB, or large-scale analytics using PostgreSQL or Cassandra, IBM Power provides the horsepower, scalability, and flexibility to support both the past and the future of enterprise data management.

Deployment options on IBM Power are diverse, ranging from traditional on-premises setups on AIX, IBM i, or Linux, to virtualized environments via PowerVM or IBM's PowerVS cloud. Additionally, containerized deployments using Red Hat OpenShift on Power are gaining traction, offering a cloud-native path to run modern databases like MongoDB and PostgreSQL efficiently to support both the past and the future of enterprise data management.

Table 6-1 provides a list of many of the databases available on IBM power. It includes information on the supported environments and provides notes on 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, RPG/COBOL systems	Native on IBM i	Deep OS integration
IBM Db2 LUW	Relational (SQL)	AIX, Linux	Analytics, data warehousing, enterprise SQL	Native or Container	Supports JSON, REST
PostgreSQL	Relational (SQL)	Linux	Modern apps, analytics, microservices	Native or Container	Extensible (e.g., JSONB, GIS)
MySQL/Maria DB	Relational (SQL)	Linux	Web backends, reporting	Native or Container	Lightweight and widely used
MongoDB	NoSQL (Document)	Linux	Unstructured data, rapid dev, IoT	Native or Container	JSON-like docs, schema-less
Cassandra	NoSQL (Wide Column)	Linux	Big data, time-series, distributed systems	Native or Container	High availability
Redis	NoSQL (Key-Value)	Linux	Caching, real-time processing	Native or Container	In-memory, fast I/O
CockroachDB	Distributed SQL	Linux (OpenShift)	Cloud-native SQL, fault-tolerant apps	(via OpenShift)	Scales like NoSQL, behaves like SQL
TimescaleDB	Time-Series (SQL)	Linux (Postgres ext)	Monitoring, IoT, event data	(PostgreSQL-based)	Uses Postgres as backend
etcd	Key-Value (Config)	Linux (K8s)	Cluster state/config in Kubernetes	(Kubernetes env)	Used in OpenShift and K8s

We provide additional information on databases in A.3.4, “Databases” on page 370

6.4 CI/CD tools

CI/CD, which stands for Continuous Integration and Continuous Delivery (or Deployment), is a modern DevOps practice that automates the process of building, testing, and deploying software. In the CI phase, developers frequently commit code changes to a shared repository, triggering automated builds and tests to ensure the new code integrates smoothly with the existing system. This rapid feedback loop helps identify bugs early, encourages collaboration, and ensures higher code quality. Popular tools used for CI include Jenkins, GitHub Actions, and GitLab CI, all of which can be integrated with source control platforms and container environments.

The CD phase automates the delivery or deployment of the tested code to production or staging environments. This allows organizations to release new features, updates, or patches more frequently and reliably, reducing manual intervention and deployment risks. On IBM Power Systems, especially in hybrid environments with IBM i, AIX, or Linux, CI/CD enables seamless modernization efforts—integrating legacy systems with microservices, APIs, and containerized workloads. By using pipelines that span tools like MERLIN, RDi, OpenShift, and Ansible, enterprises can adopt agile development practices while maintaining the reliability of their core applications.

6.4.1 Github

GitHub is a web-based interface that uses Git, the open source version control software that lets multiple people make separate changes to software at the same time. GitHub is a website that encourages collaboration and provides most of the open source software (OSS). To modernize your applications on IBM POWER using GitHub, lets discuss some new features in Github to utilize:

GitHub Copilot is an AI-powered coding assistant developed collaboratively by GitHub and OpenAI. It's built on the foundation of the GPT (Generative Pre-trained Transformer) technology, which enables it to understand natural language descriptions and generate code based on context.

Unlike traditional code auto-complete tools (such as those found in common code editors like VSCode), Copilot goes beyond simple suggestions by offering complete lines, functions, or even entire code blocks based on your coding context and intentions.

Github provides various ways to test OSS, including Continuous Integration (CI), Continuous Deployment or Delivery (CD) via Github Actions, Travis, or Jenkins.

Github Provides hosting for binary or package builds for a project. Additionally, you can utilize the Wiki page on Github to provide instructions about where to download the project for IBM POWER.

6.4.2 OpenShift Pipelines

OpenShift Pipelines is a Kubernetes-native CI/CD solution based on the open-source Tekton project, designed to automate the building, testing, and deployment of applications in a containerized environment. Fully integrated into Red Hat OpenShift, it allows developers to define and run pipelines as Kubernetes resources using YAML, enabling seamless orchestration of CI/CD workflows alongside application workloads. Each step in a pipeline runs in its own container, making pipelines highly scalable, portable, and cloud-native. Developers can build reusable tasks, trigger pipelines via Git events or image changes, and leverage built-in integrations with Git repositories, container registries, and deployment tools.

By using OpenShift Pipelines, organizations gain the ability to implement DevOps practices with greater consistency, traceability, and automation across hybrid and multi-cloud environments. It empowers teams to shift left in the development cycle, catching issues early while accelerating delivery. Combined with OpenShift GitOps and tools like Red Hat Developer Hub or IBM’s DevOps for IBM i, pipelines enable a unified CI/CD approach that supports both modern microservices and legacy workloads running on IBM Power Systems. This results in faster, more reliable software releases with reduced manual intervention and better alignment between development and operations teams.

6.4.3 OpenShift GitOps

OpenShift GitOps is Red Hat’s enterprise-grade implementation of GitOps, a powerful operational model that uses Git as the single source of truth for defining and managing infrastructure and application configurations. Built on top of the open-source Argo CD project, OpenShift GitOps enables you to automatically sync your OpenShift (or Kubernetes) cluster state with what is defined in your Git repositories. This means any changes to applications, configurations, secrets, or even infrastructure are version-controlled, auditable, and automatically deployed – all driven by Git commits.

For teams building on IBM Power Systems – whether on Linux, IBM i, or AIX (with containerized adapters) – OpenShift GitOps brings consistency and control across hybrid environments. It supports multi-cluster management, making it easier to manage applications across on-premises, edge, and public cloud deployments. By combining GitOps with tools like Red Hat OpenShift, Ansible Automation Platform, and IBM Cloud Pak solutions, organizations gain a modern, secure, and scalable approach to software delivery—automating not just application deployments but also infrastructure provisioning, configuration, and compliance.

6.5 Support for .NET on IBM Power

Initially developed to run on the x86 platform, .NET has become a cornerstone of modern software development for several compelling reasons.

- Its versatility allows developers to build a wide array of applications, from web and mobile to desktop and cloud-based solutions, even extending to areas like gaming and IoT. This broad applicability makes it a valuable skill set and a powerful tool for diverse projects.
- In addition, .NET boasts a rich ecosystem with extensive libraries, tools, and frameworks (like ASP.NET Core for web development and Xamarin for mobile), significantly accelerating the development process and providing solutions for common tasks. The integrated development environment, Visual Studio, further enhances developer productivity with its comprehensive features.
- .NET's commitment to cross-platform compatibility (through .NET Core and subsequent versions) has broadened its reach beyond Windows, enabling applications to run on macOS and Linux (including Linux on IBM Power). This flexibility is crucial in today's diverse technological landscape.
- Performance is another key factor, with .NET applications known for their speed and efficiency, often leveraging features like Just-In-Time (JIT) and Ahead-of-Time (AOT) compilation.
- Security is also paramount, and .NET incorporates robust built-in security features to protect applications from common threats.

.NET's presence on IBM Power Systems is becoming increasingly significant, providing compelling opportunities for application modernization, workload consolidation, and enhanced performance. Thanks to the open-source nature of .NET and active collaboration between IBM and Red Hat, .NET Core and subsequent versions are fully supported on the Power architecture (ppc64le), readily available through standard Linux package managers and container registries. While Microsoft's direct commercial support doesn't extend to IBM Power binaries, the robust support provided by Red Hat ensures timely updates and a stable environment for running .NET applications on Linux distributions like RHEL.

NTi Data Provider is a native .NET solution that enables efficient and secure communication between .NET applications and IBM i systems on IBM Power. Unlike generic drivers, NTi offers optimized performance by directly interacting with IBM i resources like DB2®

databases, CL commands, and programs. Its native .NET architecture ensures cross-platform compatibility and eliminates the need for special IBM i installations. By simplifying integration and often leveraging secure internal networks on Power systems, NTi facilitates the modernization of IBM i applications by allowing .NET developers to seamlessly access and extend existing business logic with modern .NET capabilities, ultimately providing a faster, more secure, and developer-friendly integration experience.

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, ensuring high availability, scalability, and fault tolerance.

One of the key features of Kubernetes is its ability to abstract 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. This 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 desired state of the system is defined in configuration files (typically YAML or JSON). These files specify the containers to be run, their resources, networking, and other parameters, and Kubernetes ensures that the system matches this desired state. It 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 has become the standard for container orchestration and is widely adopted across industries for building and managing cloud-native applications.

We discuss Kubernetes in more detail in section 10.2.2, “Kubernetes” on page 334.

6.6.1 OpenShift

Red Hat OpenShift is the leading hybrid cloud application platform, bringing together a comprehensive set of tools and services that streamline the entire application lifecycle. Red Hat OpenShift allows organizations to operate consistently across any infrastructure with full-stack automated operations.

Red Hat OpenShift is powered by the open source project Kubernetes. Kubernetes combines running a containerized infrastructure with production workloads utilizing Docker container management tools. The Kubernetes infrastructure provides an isolated and secure app platform for managing containers that is portable, extensible, and self-healing in case of a failover.

Red Hat OpenShift lets organizations accelerate their cloud-native journey and enables them to build new cloud-native, containerized applications, while benefiting from IBM Power features. Available in self-managed or fully managed cloud service editions, Red Hat OpenShift offers a complete set of integrated tools and services for cloud-native, AI, virtual, and traditional workloads alike.

In February 2025 Version 4.18 of Red Hat OpenShift became Generally Available (GA). Utilizing the features below is one way to modernize applications for IBM hardware.

Notable latest enhancements to utilize for IBM Power include:

- Multiple Architecture Cluster support
- Multiarch Tuning Operator
- Secondary Scheduler Operator
- Tuning etcd latency tolerances
- Installer Provisioned Infrastructure for IBM PowerVS - move to CAPI
- Adding compute nodes to on-premise clusters using OpenShift CLI (oc)

Red Hat OpenShift is discussed in more detail in section 10.2.1, “OpenShift” on page 332.

6.6.2 Cloud Paks

IBM Cloud Paks are a suite of pre-integrated, containerized software solutions that run on Red Hat OpenShift and are designed to help businesses quickly modernize applications, automate operations, manage data, and implement AI across hybrid cloud environments. Each Cloud Pak includes IBM middleware, open-source components, Kubernetes operators, and certified security – providing enterprises with a consistent, scalable foundation for digital innovation.

There are several IBM Cloud Paks, each focused on a specific domain:

- Cloud Pak for Applications
- Cloud Pak for Data
- Cloud Pak for Integration
- Cloud Pak for AIops

IBM Cloud Pak for Applications

IBM Cloud Pak for Applications is an enterprise-ready, containerized software solution designed to 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 help organizations transition between public, private, and hybrid clouds.

IBM Cloud Pak for Applications includes IBM Cloud Transformation Advisor, an AI-powered tool which assists in refactoring and rearchitecting legacy applications. The solution includes automated vulnerability assessment and identification, ensuring continuous security compliance across all deployment environments. It also automates audit reporting, simplifying compliance management. Developers can use their preferred IDEs to build and deploy applications, with support for modern runtimes and DevOps workflows. This integration streamlines the development process and enhances productivity.

IBM Cloud Pak for Data

IBM Cloud Pak for Data allows you to unify and simplify data collection, organization, and analysis. It is ideal for AI and analytics workloads. IBM Cloud Pak for Data is a unified, pre-integrated data and AI platform designed to help organizations collect, organize, analyze, and infuse AI into their data. Running natively on the Red Hat OpenShift Container Platform, it supports deployment across various cloud environments, including IBM Cloud, Amazon Web Services (AWS), and Microsoft Azure.

The platform allows secure access to data at its source, eliminating the need for data migration and reducing data silos, ensuring seamless data integration. It creates a trusted, business-ready analytics foundation, simplifying data preparation, policy enforcement, security, and compliance, while automating data governance and the AI lifecycle. IBM Cloud Pak for Data provides tools for building, deploying, and managing AI and machine learning models, scaling these capabilities consistently across the organization to enable comprehensive data analysis and insights.

By operationalizing AI throughout the business with trust and transparency, the platform supports the end-to-end AI workflow, ensuring effective integration of AI into business processes. Offering a single interface for end-to-end analytics with built-in governance, IBM Cloud Pak for Data simplifies the management of data and AI capabilities, while its scalable Kubernetes environment allows organizations to grow their data and AI capabilities as needed. Supporting multi-cloud deployments, it provides agility and avoids vendor lock-in, making it a powerful tool for accelerating the journey to AI and unlocking the value of data for AI-driven digital transformation.

IBM Cloud Pak for Integration

IBM Cloud Pak for Integration is a comprehensive, AI-powered hybrid integration platform designed to connect applications, data, systems, and services across any environment. It provides a unified experience with a suite of integration tools that streamline the creation, management, and deployment of integrations. Running on Red Hat OpenShift, IBM Cloud Pak for Integration supports both cloud and on-premises deployments, ensuring scalability and security. The platform includes components such as IBM API Connect® for managing APIs, IBM App Connect for no-code integration, and IBM Event Streams for real-time data processing. By leveraging AI and automation, IBM Cloud Pak for Integration accelerates the integration process, reduces manual workflows, and enhances responsiveness to real-time events. This makes it an ideal solution for organizations looking to modernize their integration capabilities and drive digital transformation.

IBM Cloud Pak for Business Automation

IBM Cloud Pak for Business Automation is a modular set of integrated software components designed to automate work and accelerate business growth. Built for any hybrid cloud, it simplifies complex workflows, facilitates records management, and enhances overall productivity. The platform uses AI to identify gaps and build low-code and no-code automations, making it easier to streamline operations. Running on Red Hat OpenShift, IBM Cloud Pak for Business Automation supports containerized deployments across various cloud environments, providing flexibility and scalability. Key features include automating case and process workflows, converting unstructured content into valuable data, and using software robots to complete tasks based on AI insights. This comprehensive automation solution helps organizations improve efficiency, reduce operational costs, and drive continuous process improvements.

IBM Cloud Pak for AIOps

IBM Cloud Pak for AIOps is an advanced, AI-powered platform designed to enhance IT operations (ITOps) by leveraging artificial intelligence and machine learning. It integrates seamlessly with existing ITOps toolchains to provide comprehensive visibility, proactive incident management, and automated remediation. By analyzing data from various sources, such as logs, metrics, and events, IBM Cloud Pak for AIOps helps IT teams predict and resolve issues before they impact business operations. The platform supports hybrid cloud environments, enabling organizations to manage their IT infrastructure across on-premises, cloud, and containerized environments. Key features include event correlation and compression, anomaly detection, root cause analysis, and automated runbooks, all aimed at reducing mean time to resolution (MTTR) and improving overall operational efficiency. With its collaborative tools and real-time insights, IBM Cloud Pak for AIOps empowers IT teams to innovate faster, reduce operational costs, and ensure the reliability of mission-critical workloads.

7

Tools and Performance

Modernized application environments, characterized by cloud-native architectures, microservices, containerization, and DevOps practices, present a unique set of challenges for tools and performance management. These environments are dynamic, distributed, and complex, making traditional monitoring and management approaches inadequate.

Modern applications often rely on microservices, which can be spread across different servers or even clouds. This increases complexity as tools need to track interactions across various services, each potentially written in different languages and using different technologies. Tools must be able to track the dynamic nature of modern applications, where services may scale up or down, and the infrastructure is continuously changing.

Traditional performance monitoring may focus only on specific parts of the stack (e.g., CPU usage, memory consumption, or database performance), but modern applications require an end-to-end view of everything from frontend performance to backend infrastructure. Getting a complete picture across distributed environments can be challenging. In modernized applications, there’s an enormous volume of performance data generated, from logs and metrics to traces and events. Collecting, storing, and analyzing this data can overwhelm performance monitoring systems, making it harder to derive meaningful insights. Tools must be able to distinguish between normal behavior and actual issues. Filtering out the noise from irrelevant or redundant data is crucial, but this can often result in missed signals if not done properly.

Many modern applications rely on auto-scaling to adjust resource allocation dynamically based on demand. Tools need to manage and predict resource consumption effectively to prevent issues like resource contention or over-provisioning. As applications scale up or down, so do the costs, especially in cloud environments. Managing resource utilization and costs becomes a balancing act, as under-utilization may lead to inefficiencies, while over-provisioning may result in unnecessary expenses.

As AI and machine learning (ML) play an increasing role in application management, these tools need to be able to predict potential performance degradation before it happens. This requires access to massive datasets and the ability to process them effectively, as well as trust in the algorithms' accuracy. There is an increasing shift toward autonomous performance management systems that can detect and correct issues automatically. However, these

© Copyright IBM Corp. 2025.

systems are still evolving and might not always act correctly in complex or unanticipated scenarios.

The challenges of tools and performance management in modernized application environments stem largely from the complexity, scale, and rapid pace of change in today's software ecosystems. To succeed, organizations need a cohesive strategy for selecting, integrating, and optimizing their performance management tools. This strategy should focus on automation, real-time monitoring, and cross-team collaboration to stay on top of performance issues while maintaining a smooth and responsive user experience.

The following topics are covered in this chapter:

- 7.1, “Monitoring” on page 199
- 7.2, “Logging” on page 210
- 7.3, “Autoscaling and Quality of Service” on page 214
- 7.4, “Management” on page 215

7.1 Monitoring

Monitoring is foundational to any successful application modernization initiative. Before a single line of code is refactored or a new service deployed, establishing a robust monitoring framework is paramount. This initial phase involves meticulously capturing the current state of the application's performance. Metrics such as response times for critical user flows, error rates across different components, the consumption of underlying infrastructure resources like CPU, memory, and network bandwidth, and even end-user experience metrics become crucial benchmarks. These baselines provide an objective understanding of the “before” picture, allowing teams to set tangible goals for improvement in the modernized application. Without this initial visibility, it becomes exceedingly difficult to measure the impact of modernization efforts or identify regressions.

As the modernization journey progresses, whether through incremental refactoring, replatforming to cloud environments, or a complete re-architecture using microservices, continuous monitoring acts as a vital compass. It provides real-time visibility into the health and performance of the application during each stage of the transformation. Deploying a new version of a service, migrating a database, or integrating a new API can introduce unforeseen issues. Monitoring tools immediately flag any deviations from expected behavior, such as increased latency, elevated error counts, or resource contention. This early detection mechanism empowers development and operations teams to address problems proactively, preventing minor glitches from snowballing into significant outages or performance degradation that could impact end-users. Furthermore, monitoring serves as a critical validation step after each deployment, confirming that the modernized components are functioning correctly within the new environment and interacting seamlessly with existing parts of the system.

Once the application has been fully modernized and deployed in its target environment, the role of monitoring evolves but remains equally critical. The new architecture, often involving distributed systems and cloud-native technologies, introduces its own set of complexities. Monitoring now focuses on identifying performance bottlenecks that might emerge in this new landscape, such as inefficient inter-service communication, suboptimal database queries, or limitations in cloud resource provisioning. By continuously analyzing resource utilization patterns, teams can optimize resource allocation, potentially leading to significant cost savings in cloud environments. Moreover, monitoring plays a key role in ensuring the scalability of the modernized application. Observing performance under varying load conditions helps identify areas that need further optimization to handle increased user demand without compromising responsiveness or stability.

Beyond performance, monitoring is indispensable for maintaining the reliability and stability of the modernized application. It continuously tracks uptime and availability, ensuring adherence to defined service level agreements (SLAs). Comprehensive error tracking and logging capabilities provide the detailed information needed to diagnose the root cause of any failures, facilitating faster resolution and minimizing downtime. Proactive alerting, configured based on key performance indicators, acts as an early warning system, notifying teams of potential issues before they escalate and impact users. This proactive approach shifts the focus from reactive incident management to preventative measures, enhancing the overall resilience of the application.

A crucial aspect of the user-centric approach in modern application development is understanding the actual end-user experience. Real User Monitoring (RUM) tools, integrated into the monitoring strategy, provide invaluable insights into how users are interacting with the modernized application. Metrics such as page load times, the occurrence of JavaScript errors, and overall responsiveness as perceived by users are captured and analyzed. This data helps identify specific areas of the application where users might be encountering friction or frustration, guiding optimization efforts to enhance user satisfaction and improve the overall user journey.

Finally, monitoring provides the data-driven foundation for continuous improvement and iterative development. The wealth of information collected on performance, reliability, and user experience informs future development decisions. After implementing performance enhancements or introducing new features, monitoring allows teams to validate their effectiveness and identify any unintended consequences. This continuous feedback loop, powered by comprehensive monitoring, ensures that the modernized application remains performant, reliable, and aligned with evolving user needs and business objectives. In essence, monitoring is not just a technical necessity; it is a strategic imperative for realizing the full benefits of application modernization.

7.1.1 Monitoring Solutions

Monitoring solutions are integral to modern infrastructure, providing real-time insights and proactive management capabilities that ensure optimal performance, security, and reliability. In this section we discuss just some of the solutions available to you.

Instana is a powerful observability platform that leverages AI and automation to monitor applications across cloud-native environments. It offers high-fidelity data updated every second, enabling quick identification and resolution of issues before they impact users 1. Instana's comprehensive monitoring capabilities cover everything from microservices to Kubernetes, providing a unified view of the entire application stack.

Turbonomic is another advanced monitoring solution that focuses on application resource management. It dynamically allocates resources in real-time to optimize performance and reduce costs. Turbonomic's intelligent automation and proactive resource management ensure continuous application performance across hybrid and multicloud environments.

Additionally, the OpenShift monitoring stack integrates tools like Prometheus, Grafana, and Alertmanager to provide detailed metrics, visualization, and alerting capabilities for containerized applications. This stack ensures that both platform components and user workloads are monitored effectively, supporting the health and efficiency of modern IT environments.

IBM Instana Observability

IBM Instana Observability is a comprehensive, fully automated enterprise observability platform designed for modern, dynamic applications, including those built with microservices, containers (like Kubernetes and Docker), and cloud-native architectures. It provides the contextual insights needed for teams to take intelligent actions and ensure optimal application performance.

A key differentiator of Instana is its automatic discovery and monitoring of the entire application stack and underlying infrastructure. Once the Instana agent is deployed, it automatically detects and starts monitoring over 300 technologies without requiring extensive manual configuration. This includes application runtimes (such as Java, Node.js, Python), databases, middleware, and orchestration platforms. Instana then dynamically builds a real-time dependency map, visualizing the interconnectedness of all services and infrastructure components.

Core capabilities of IBM Instana Observability include full-stack visibility, offering insights from end-user interactions down to the code level and infrastructure. It captures 100% of requests and traces at a granular one-second level without sampling, ensuring no blind spots in performance monitoring. The platform leverages AI-powered analytics for automatic root cause analysis, helping teams quickly pinpoint the exact cause of performance issues and reduce mean time to resolution (MTTR). Other notable features include anomaly detection for proactive issue identification, end-user experience monitoring (EUM) for understanding user impact, infrastructure monitoring, log management in context, synthetic monitoring for proactive testing, and customizable dashboards for visualizing key metrics and creating application perspectives. IBM also offers Instana Observability on z/OS® to extend these capabilities to mainframe environments, providing end-to-end visibility in hybrid applications.

IBM Instana Observability offers different pricing tiers, such as Essentials (primarily infrastructure monitoring) and Standard (full-stack observability including APM, tracing, and more). Pricing is typically based on the number of Managed Virtual Servers (MVS) or hosts being monitored and the chosen features. There are also options for SaaS and self-hosted deployments, with self-hosted generally being more expensive. Additional costs may apply for features like synthetic monitoring from IBM-managed Points of Presence (PoPs) and extended log retention. IBM provides a pricing calculator for estimates, and private offers or free trials may be available through their website or cloud marketplaces.

Extensive documentation for Instana can be found at
https://www.ibm.com/docs/en/instana-observability/current.

IBM Turbonomic

IBM Turbonomic stands as an Application Resource Management (ARM) platform, leveraging the power of artificial intelligence to continuously optimize the performance and cost efficiency of applications operating across diverse hybrid and multi-cloud landscapes. Its primary objective is to ensure that applications consistently have the precise resources they require to function optimally, all while diligently minimizing the expenditure on underlying infrastructure. This is achieved through a dynamic and real-time matching of application demand with the available infrastructure supply.

At the heart of Turbonomic's operation lies its unique approach of modeling the IT environment as a dynamic marketplace. Within this model, applications are positioned as consumers actively seeking resources such as CPU, memory, storage, and network bandwidth. Conversely, the infrastructure acts as the provider, offering these resources for consumption. The platform employs a sophisticated economic scheduling engine that meticulously analyzes real-time resource consumption patterns and associated pricing (represented in a virtual currency). This continuous analysis empowers Turbonomic to make intelligent and automated decisions regarding resource allocation, rightsizing of instances, and optimal workload placement, ultimately striving to maintain a delicate balance between application performance and infrastructure cost.

The capabilities of IBM Turbonomic are extensive and designed to provide comprehensive control over resource management. Full-stack visibility is a cornerstone, with the platform automatically discovering and mapping the intricate relationships within the application and infrastructure stack, extending from the application layer down to the physical hardware. This holistic view provides critical context for understanding resource dependencies and potential bottlenecks. Furthermore, Turbonomic harnesses the power of AI and machine learning to analyze vast amounts of real-time data and predict future resource needs. This proactive approach allows the platform to identify potential performance issues and cost inefficiencies before they impact application availability or budgets.

A significant strength of Turbonomic lies in its ability to automate resource optimization. The platform generates actionable recommendations based on its analysis and can be configured to automatically execute these actions. This includes tasks such as dynamically resizing virtual machines, intelligently migrating workloads between different hosts or cloud regions, adjusting cloud instance types to better match demand, and scaling Kubernetes resources up or down as needed. This level of automation reduces the need for manual intervention and ensures continuous optimization. Moreover, Turbonomic plays a crucial role in cloud cost optimization by identifying and addressing over-provisioned resources, strategically leveraging reserved instances and savings plans offered by cloud providers, and recommending the most cost-effective instance types for specific workloads.

Turbonomic is also instrumental 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 multi-cloud environments is a key advantage, allowing organizations to manage resources seamlessly across their on-premises data centers (supporting technologies like VMware and hyperconverged infrastructure) and major public cloud providers (AWS, Azure, and Google Cloud). For organizations embracing containerization, Turbonomic offers specialized Kubernetes optimization capabilities, including intelligent container resizing, optimal pod placement within clusters, and dynamic cluster scaling to enhance performance and reduce costs in these complex environments.

Beyond real-time management, Turbonomic provides valuable what-if planning capabilities. Users can simulate various scenarios, such as capacity planning exercises, cloud migration strategies, and infrastructure upgrades, to understand the potential impact on both application performance and infrastructure costs before making significant changes. Finally, Turbonomic boasts a robust integration ecosystem, seamlessly connecting with a wide array of existing IT management and DevOps tools, including monitoring solutions, cloud management platforms, and automation frameworks, to provide a unified and streamlined operational experience.

Documentation for Turbonomic can be found at `https://www.ibm.com/docs/en/tarm`

Red Hat OpenShift Monitoring Stack

Red Hat OpenShift provides a robust and comprehensive monitoring stack designed to ensure the health and performance of containerized applications and infrastructure. This stack includes several key tools and components that work together to provide detailed insights and proactive management capabilities. Figure 7-1 shows a high level overview of the monitoring stack.

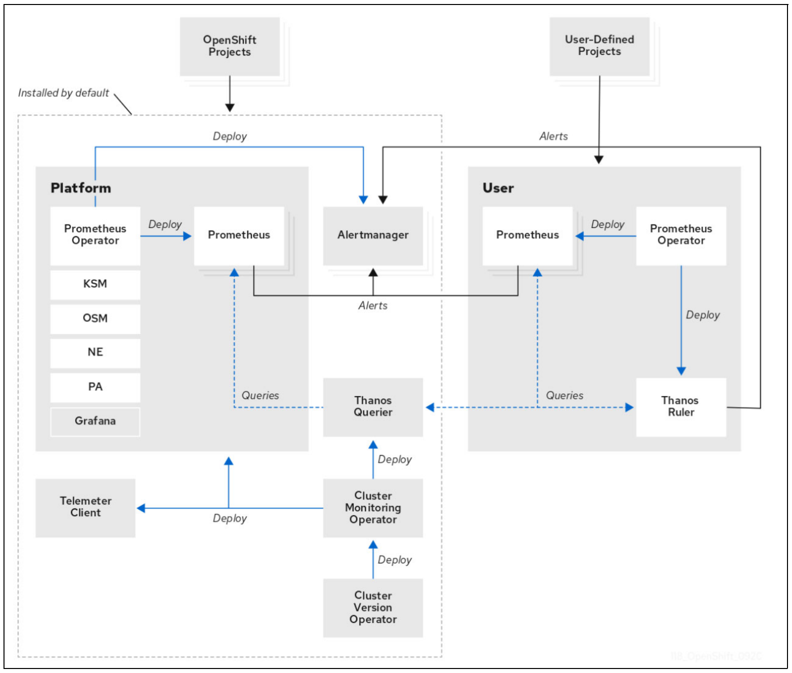


Figure 7-1 Overview of the Red Hat OpenShift monitoring stack

Prometheus

Prometheus is a core component of the OpenShift monitoring stack. It is an open-source system monitoring and alerting toolkit that collects and stores metrics as time series data. Prometheus is highly scalable and can handle millions of time series, making it ideal for monitoring large and complex environments. It scrapes data from various endpoints and stores it in its own time series database, providing detailed metrics on system performance1.

Grafana

Grafana is used in conjunction with Prometheus to visualize the collected metrics. It is an open-source platform that allows users to create and share dashboards, providing a comprehensive view of system health and performance. Grafana supports a wide range of data sources and offers powerful querying capabilities, making it easy to analyze and interpret the data collected by Prometheus.

Alertmanager

Alertmanager is another critical component of the OpenShift monitoring stack. It handles alerts generated by Prometheus and manages their delivery. Alertmanager can route alerts to various notification channels such as email, Slack, or PagerDuty, ensuring that the right people are informed about issues promptly. It also supports alert deduplication, grouping, and silencing, helping to reduce alert fatigue and improve response times.

Sysdig

Sysdig is an additional monitoring tool that can be integrated with OpenShift. It provides deep visibility into containerized environments, offering features such as container security, compliance, and performance monitoring. Sysdig uses a kernel module to capture system calls and other OS-level events, providing detailed insights into container activity and performance.

Datadog

Datadog is a cloud-based monitoring and analytics platform that can be used with OpenShift to monitor applications, infrastructure, and logs. It offers 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 provides features such as anomaly detection, dashboards, and alerting, helping organizations maintain optimal system health.

OpenShift Web Console

The OpenShift Web Console includes built-in monitoring features that provide visual representations of cluster metrics. It offers default dashboards that help administrators quickly understand the state of their cluster. The console also includes tools for managing metrics, alerts, and monitoring dashboards, making it easy to monitor and manage the OpenShift environment.

User-Defined Monitoring

OpenShift allows cluster administrators to enable monitoring for user-defined projects. This feature provides additional monitoring components that can be configured to monitor specific services and pods within user projects. It ensures that both platform components and user workloads are monitored effectively, providing comprehensive coverage across the entire environment.

In summary, Red Hat OpenShift's monitoring stack is designed to provide detailed insights into system performance, security, and reliability. By leveraging tools like Prometheus, Grafana, Alertmanager, Sysdig, and Datadog, organizations can ensure that their containerized applications and infrastructure are running smoothly and efficiently. The built-in monitoring features of the OpenShift Web Console further enhance visibility and control, making it easier 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 instrumented for Prometheus data. The application is deployed on Red Hat OpenShift cluster running on IBM Power along with Prometheus and Grafana.

Running a Prometheus instrumented application on Red Hat OpenShift

This article makes use of a go application instrumented for Prometheus. This application is shown in this Prometheus documentation.

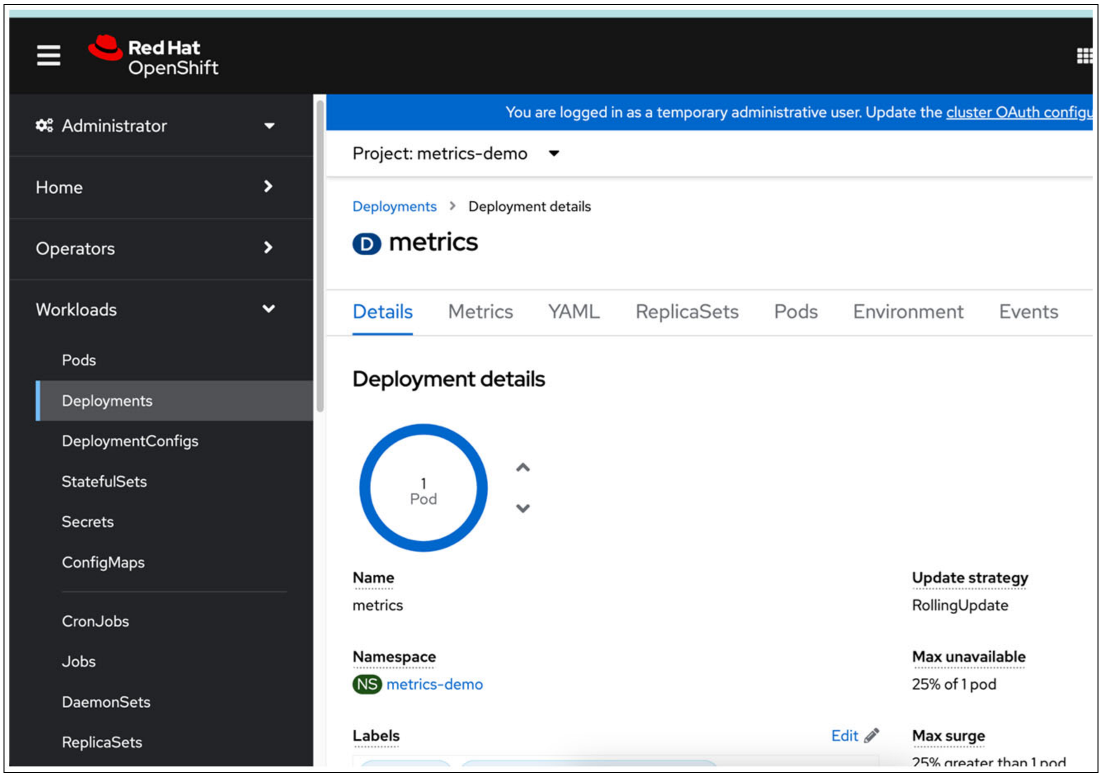
1. The Docker file to deploy the above sample application can be found at `https://github.com/mithunhr87/metrics_first/`

   An image, using above docker file, can be built and pushed to a Quay repository using the commands 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



You are logged in as a temporary administrative user. Update the cluster OAuth configu

Project: metrics-demo

D metrics

Deployment details

Name
metrics

Update strategy
RollingUpdate

Namespace
NS metrics-demo

Max unavailable
25% of 1 pod

Labels

Max surge
25% greater than 1 pod

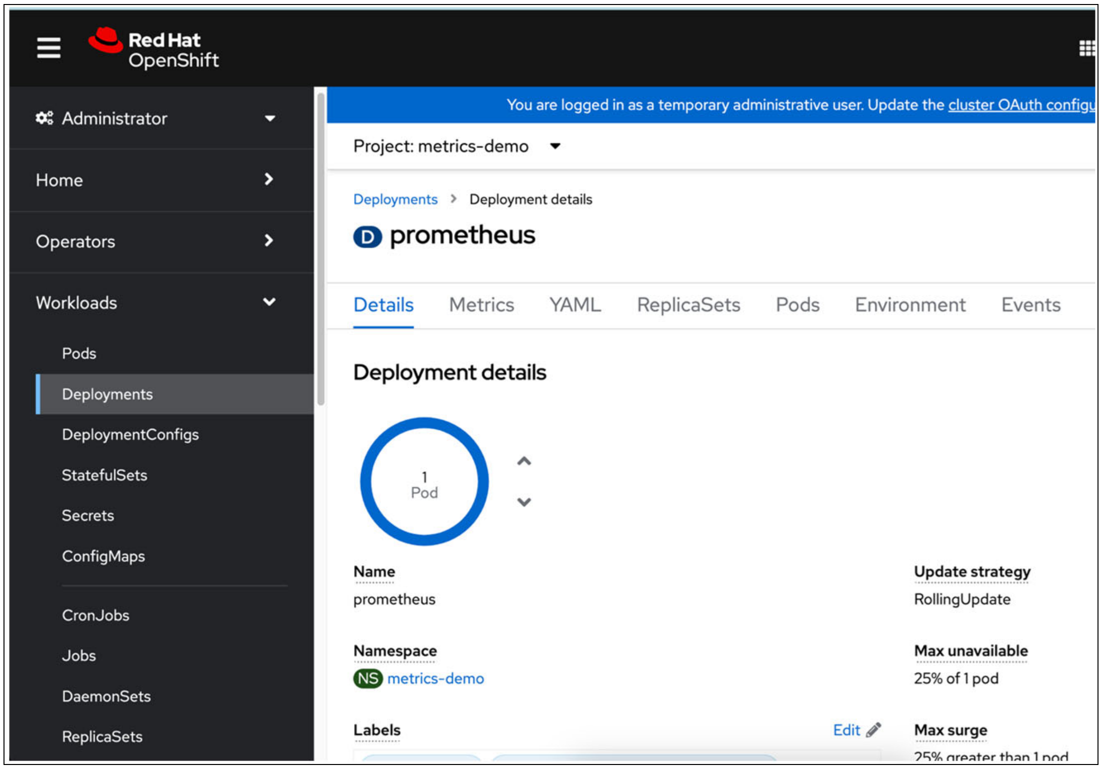
Figure 7-2 Project deployment details

2. Deploying the Prometheus instance

   Prometheus is deployed using the publicly available docker image available for IBM Linux on POWER Systems (ppc64le) at docker.io using the command:

   ```
   #docker pull prom/Prometheus
   ```

Figure 7-3 shows the Prometheus deployment.



You are logged in as a temporary administrative user. Update the cluster OAuth configu

Project: metrics-demo

prometheus

Deployment details

Name
prometheus

Namespace
NS metrics-demo

Labels

Update strategy
RollingUpdate

Max unavailable
25% of 1 pod

Max surge
25% greater than 1 pod

Figure 7-3 Prometheus deployment

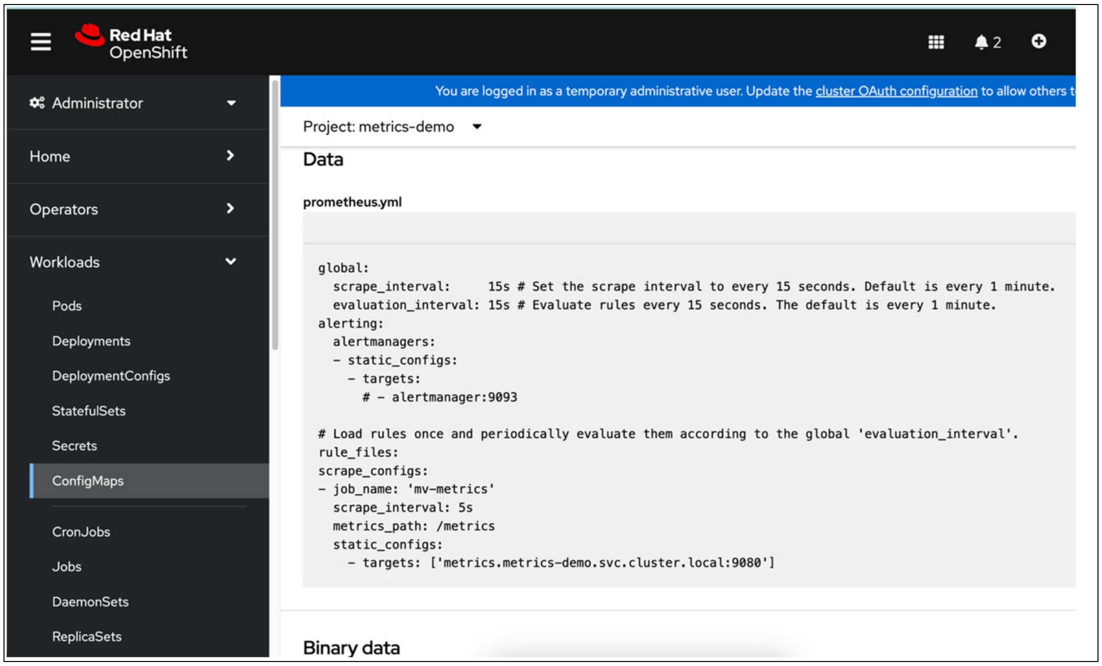
3. Create a Prometheus.yml config map containing the scrape information

   The default Prometheus setup won't automatically know where to scrape data from for our instrumented application. Therefore, we need to create a configuration map that includes these details. This is done in the promotheus.yml file shown in Example 7-2.

Example 7-2 Prometueus.yml file

```
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']
```

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
https://github.com/mithunhr87/metrics_first/blob/main/prometheus.yml

4. In the default Prometheus instance, create volumes and volume mounts. Then, load the prometheus.yml file and force it to read the updated prometheus.yml file

   a. Under *volumes* in Prometheus deployment file add the lines 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 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 shown in Example 7-5 to image to read the new Prometheus file.

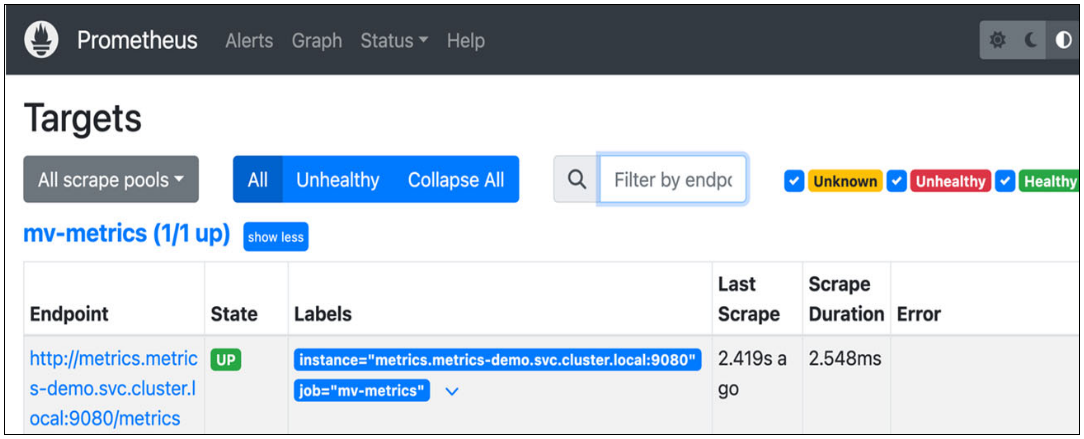
Example 7-5 Arguments for Prometheus

```
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'
```

The complete Prometheus modified deployment file can be found at
https://github.com/mithunhr87/metrics_first/blob/main/prometheus_deployment.yaml

5. Check for targets on Prometheus

   Now the Prometheus will have the target displayed for the instrumented go application as shown in Figure 7-5.



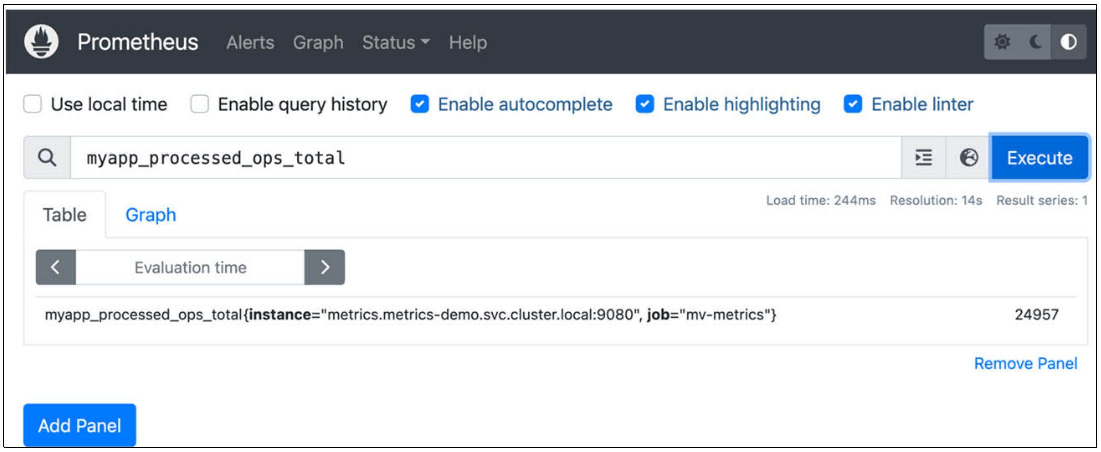
Targets

mv-metrics (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://metrics.metrics-demo.svc.cluster.local:9080/metrics	UP	instance="metrics.metrics-demo.svc.cluster.local:9080" job="mv-metrics"	2.419s ago	2.548ms	

Figure 7-5 Prometheus instance after configuration

Check for *myapp_processed_ops_total* parameter to check the total operations processed



□ Use local time □ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

```
myapp_processed_ops_total
```

Load time: 244ms Resolution: 14s Result series: 1

```
myapp_processed_ops_total{instance="metrics.metrics-demo.svc.cluster.local:9080", job="mv-metrics"}    24957
```

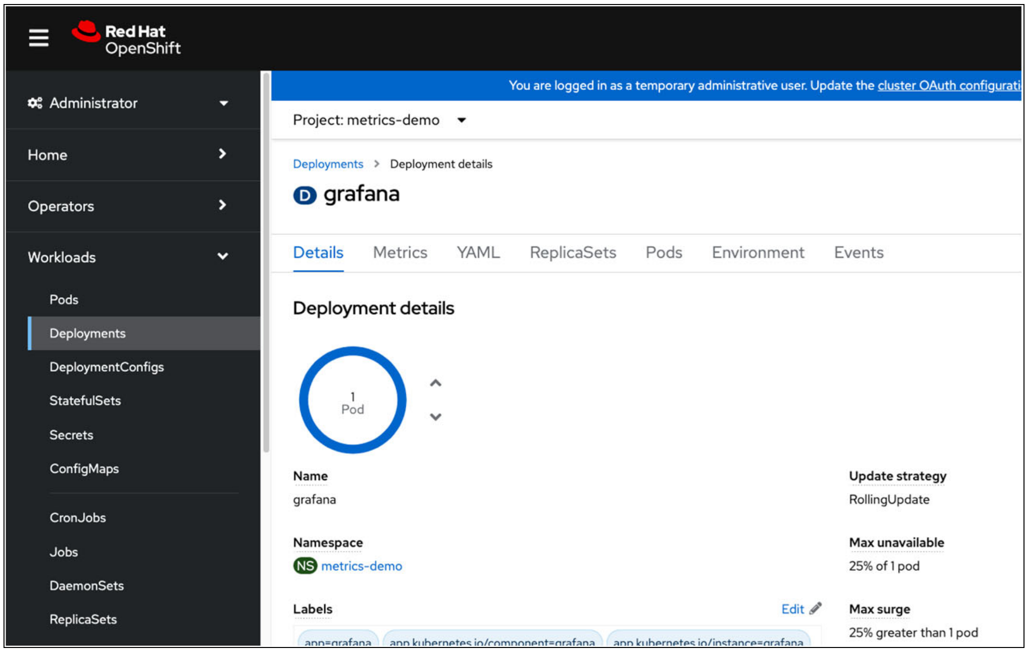
Figure 7-6 Operations processed.

6. Deploying Grafana

   The Grafana Image for IBM Linux on POWER Systems (ppc64le) can be found in the Red Hat catalog.

   a. Deploy Grafana.

      The deployment of the Grafana image on Red Hat OpenShift is shown in Figure 7-7.



You are logged in as a temporary administrative user. Update the cluster OAuth configurati

Project: metrics-demo

Deployment details

Name
grafana

Namespace
NS metrics-demo

Labels
app=grafana app.kubernetes.io/component=grafana app.kubernetes.io/instance=grafana

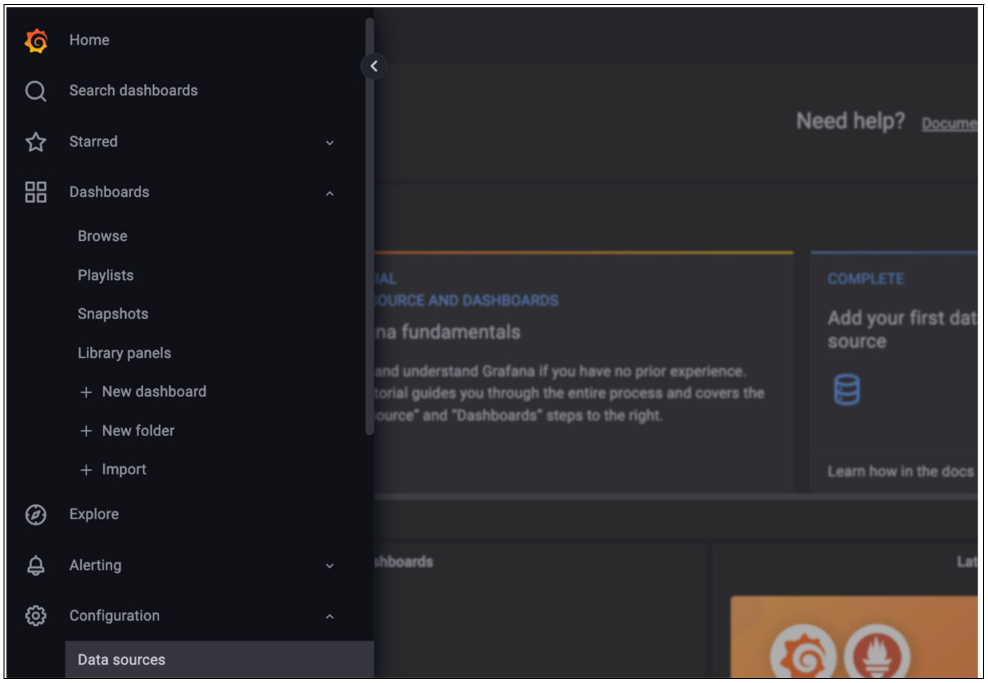
Update strategy
RollingUpdate

Max unavailable
25% of 1 pod

Max surge
25% greater than 1 pod

Figure 7-7 Grafana deployment

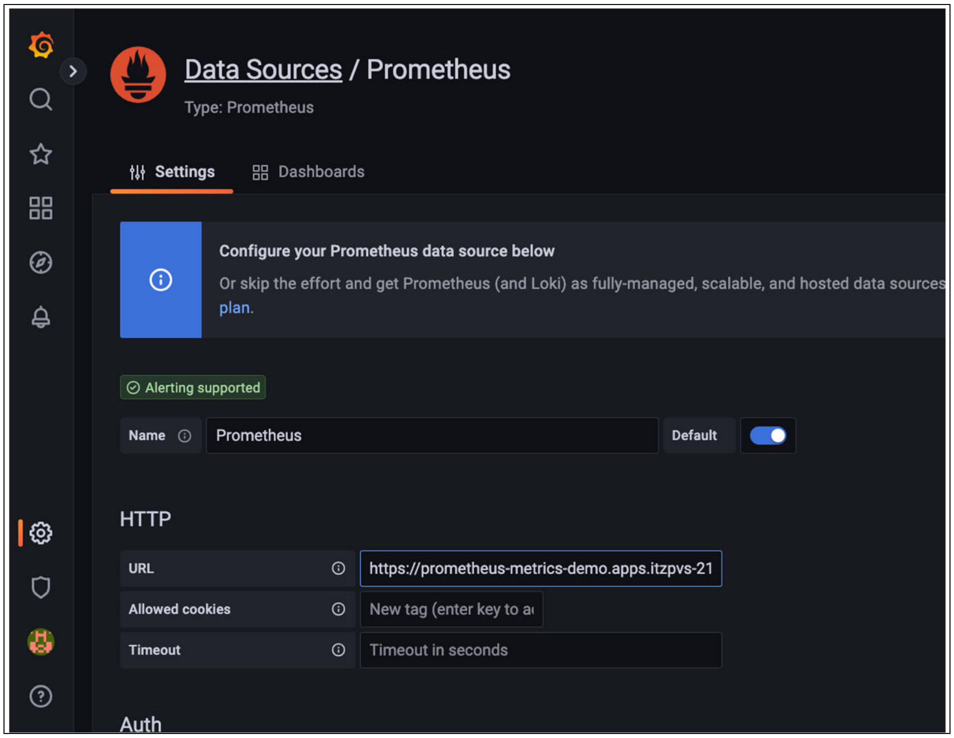
   b. Within Grafana, select data sources from the menu to add a new data source as shown in Figure 7-8.



and understand Grafana if you have no prior experience.
torial guides you through the entire process and covers the
ource" and "Dashboards" steps to the right.

Figure 7-8 Select data sources

c. Provide the URL of the Prometheus instance deployed as shown in Figure 7-9.


Data Sources / Prometheus

Type: Prometheus

Configure your Prometheus data source below

Or skip the effort and get Prometheus (and Loki) as fully-managed, scalable, and hosted data sources

plan.

Alerting supported

Name: Prometheus

Default: ●

HTTP

URL: https://prometheus-metrics-demo.apps.itzpvs-21

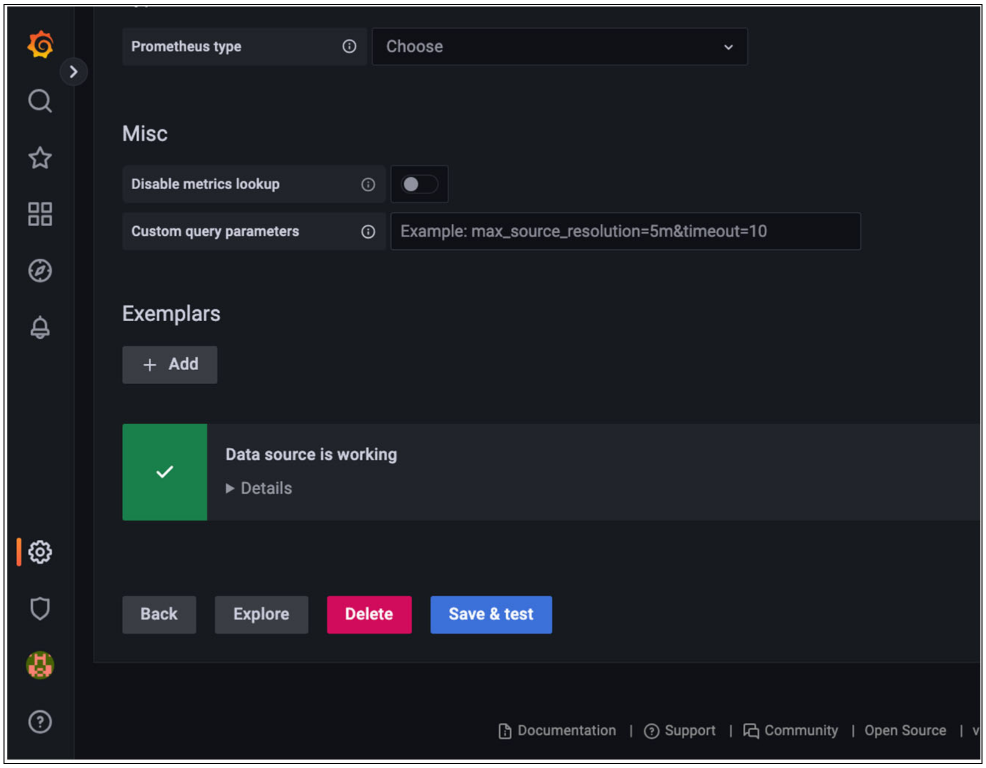
Allowed cookies: New tag (enter key to a

Timeout: Timeout in seconds

Auth

Figure 7-9 Entering URL of Prometheus instance

Select **Save and test**. If you are successful, a message “data source is working” will be displayed as shown in Figure 7-10.


Prometheus type: Choose

Misc

Disable metrics lookup: ○

Custom query parameters: Example: max_source_resolution=5m&timeout=10

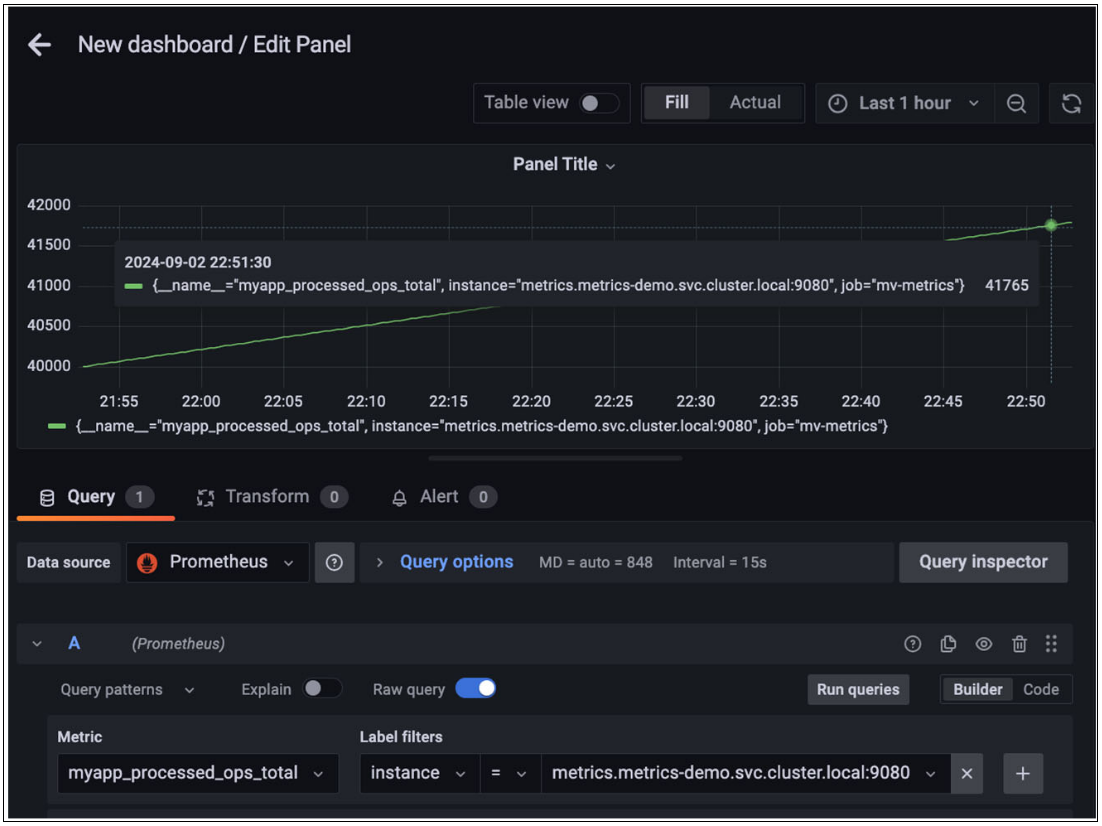
Exemplars

Data source is working

▶ Details

Figure 7-10 Successful data connection

7. Create a new dashboard and add the new panel, with data source Prometheus. You can then query the data. Figure 7-11 shows 'myapp_processed_ops_total' displayed for the past hour.



2024-09-02 22:51:30
{__name__="myapp_processed_ops_total", instance="metrics.metrics-demo.svc.cluster.local:9080", job="mv-metrics"} 41765

{__name__="myapp_processed_ops_total", instance="metrics.metrics-demo.svc.cluster.local:9080", job="mv-metrics"}

Data source: Prometheus

MD = auto = 848 Interval = 15s

Metric: myapp_processed_ops_total

Label filters: instance = metrics.metrics-demo.svc.cluster.local:9080

Figure 7-11 Dashboard showing statistics

We have now shown how to create a complete monitoring stack using a customized Prometheus configuration and using Grafana as a deployment 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 be performed at various levels, including system, server, network, and application. The events logged may be categorized by different security levels, such as trace, debug, info, warning, error, and fatal.

Integrating a logging framework during the initial stages of application design is crucial. Application logging helps track how the application behaves under different conditions and loads 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.In this section we discuss the most used logging software, components and capabilities in modern environments.

7.2.1 Traditional System Logging

AIX uses the syslog daemon or syslogd, to filter and log system events. These events can be redirected to a central logging server. In Red Hat Linux this daemon is called the rsyslogd. Traditionally, we can easily identify the messages coming from a particular VM which hosts critical applications.

In a modern environment, for OpenShift on IBM Power, how do we manage, redirect and filter logging events for a cluster of multiple nodes, running multiple containers, configured with multiple routes and services for multiple applications.

7.2.2 Container logging

Containers typically log to standard output streams, namely stdout and stderr. However, the log format varies depending on the container runtime. For Docker, the default logging format is JSON, whereas for CRI-O, which is used by OpenShift, the format is plain text. With OpenShift transitioning from Docker to CRI-O as its default container engine, not all logging tools are capable of accurately parsing and formatting all container logs. Container logging is illustrated in Figure 7-12.

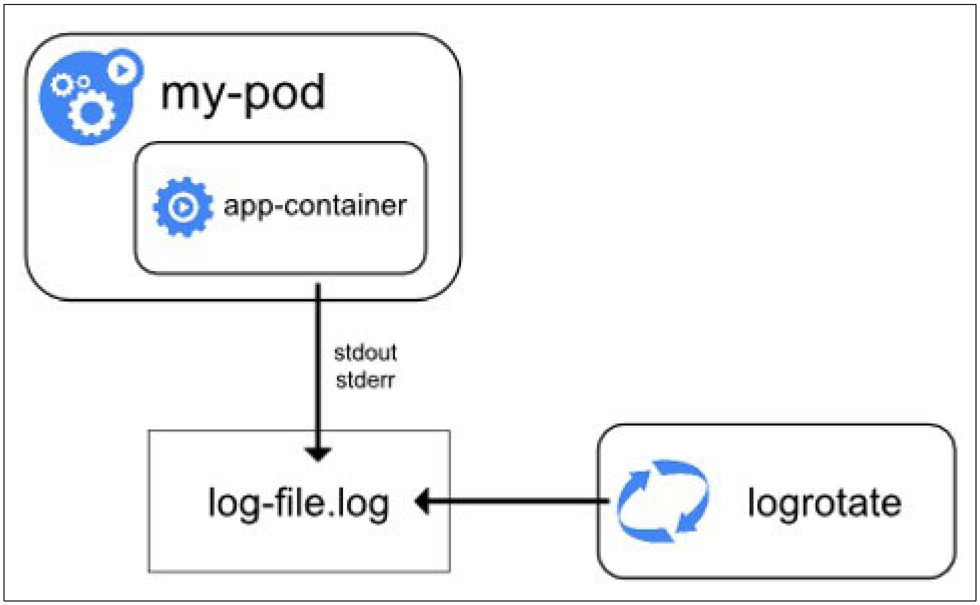


Figure 7-12 Kubernetes logging[1]

A container runtime manages and redirects any output generated by a containerized application's *stdout* and *stderr* streams. While different container runtimes implement this in various ways, their integration with the kubelet follows the standardized CRI logging format.

When a container restarts, Kubernetes retains one terminated container along with its logs. If a pod is evicted from a node and subsequently removed or deleted, all associated containers and their logs are also evicted. The kubelet makes these logs accessible to clients through a special feature of the Kubernetes API, typically accessed using kubectl logs or oc logs.

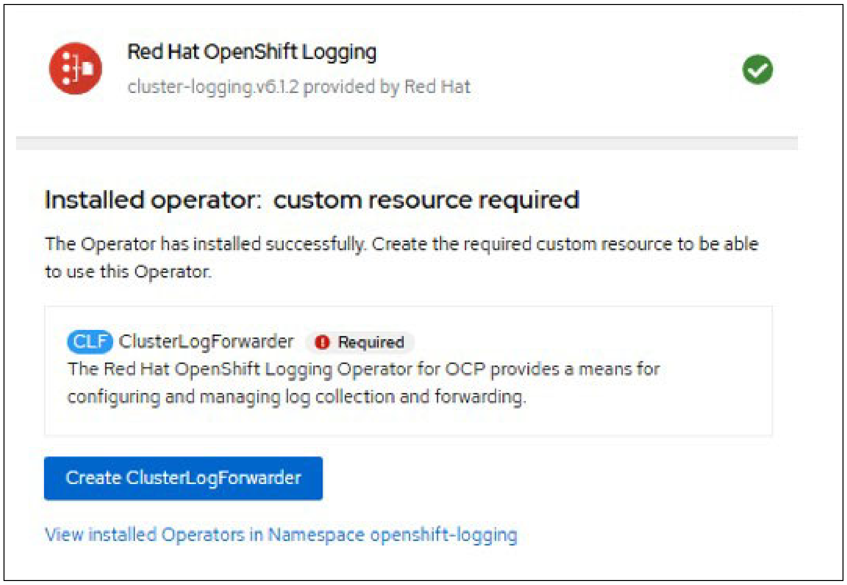
The challenge lies in persisting log data, similar to traditional syslogs, by redirecting and filtering logs to a central log server. In modern infrastructure, this central log server is often a central logging container.

[1] https://kubernetes.io/docs/concepts/cluster-administration/logging

7.2.3 Red Hat OpenShift Logging

Red Hat OpenShift logging is now termed observability. Since this is a declarative architecture, we observe what is happening.

OpenShift logging is easily installed as an operator in OpenShift on IBM Power.This is shown in Figure 7-13.



Red Hat OpenShift Logging

cluster-logging.v6.1.2 provided by Red Hat

Installed operator: custom resource required

The Operator has installed successfully. Create the required custom resource to be able to use this Operator.

CLF ClusterLogForwarder Required

The Red Hat OpenShift Logging Operator for OCP provides a means for configuring and managing log collection and forwarding.

Create ClusterLogForwarder
View installed Operators in Namespace openshift-logging

Figure 7-13 Logging operator

Since OpenShift Logging v5.6, OpenShift has introduced stream-based retention capabilities within the Loki Stack. This feature allows for the persistent storage of container logs in an Object Storage Bucket, ensuring that logging data is retained even after the container and pod are removed, as demonstrated in Figure 7-14.

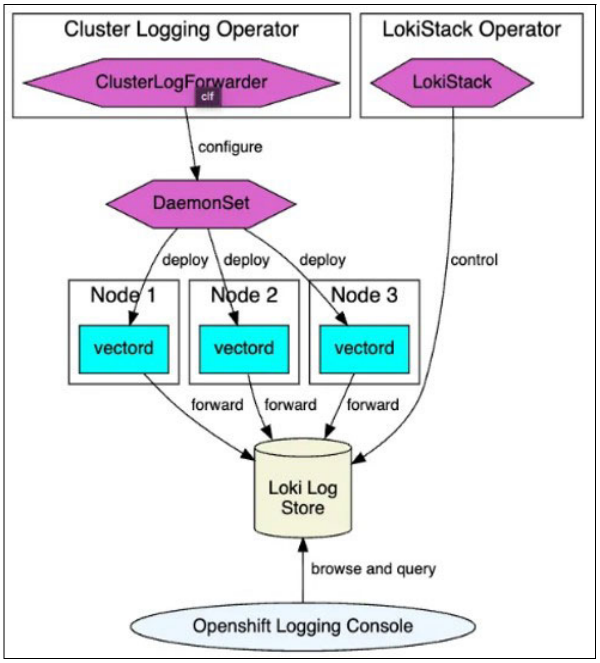


Figure 7-14 Integration of logging operator

The vector log collector is filterable and collects the log stream. While the required log data is forwarded to the persistent S3 compliant object storage in the Loki log store.

All log events are viewed in a console plugin for observability. You can also configure log forwarding from OpenShift logging to your log server.

For more information on the OpenShift logging stack refer to this Red Hat document.

Alternatively, you can configure only the collector in OpenShift logging to forward the logs to a third-party log aggregator for storage and analysis. This approach reduces the load and resource requirements on your OpenShift cluster.

The following are examples of third party logging servers that can receive logs from OpenShift logging.

- Elasticsearch
- Grafana Loki
- Splunk
- Amazon CloudWatch
- Google Cloud Logging

7.2.4 Apache Log4j

Apache Log4j is a widely-used, open-source, Java-based logging library. It provides a robust framework for recording application events, such as errors, warnings, and informational messages. These logs are crucial for debugging, monitoring application behavior, and understanding system activity. Log4j's flexibility allows developers to control which log messages are generated, their format, and where they are sent (e.g., console, files, databases).

Log4j is the latest version, offering significant improvements over its predecessor. It boasts enhanced performance, a plugin architecture for extensibility, and more flexible configuration options, supporting formats like XML, JSON, and YAML. Log4j also provides better support for asynchronous logging, which greatly improves performance in multi-threaded applications.

Key features of Log4j include:

- Asynchronous Logging: Significantly improves performance by offloading logging to a separate thread.
- Plugin Architecture: Enables customization and extension of Log4j's functionality.
- Flexible Configuration: Supports various configuration formats (XML, JSON, YAML, 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:

1. Add Log4j to your project dependencies (via Maven, Gradle, etc.).
2. Create a Logger instance in your Java classes.
3. Log messages using methods like logger.debug(), logger.info(), 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 Java applications to manage logging. It provides a robust, configurable framework to capture logs for monitoring, debugging, and error tracking. However, like any widely-used software, it is important to keep it updated, especially following vulnerabilities like Log4Shell, to maintain security and reliability in production systems.

7.3 Autoscaling and Quality of Service

A node is overcommitted when it has a pod scheduled that makes no request, or when the sum of limits across all pods on that node exceeds available machine capacity. In an overcommitted environment, it is possible that the pods on the node will attempt to use more compute resource than is available at any given point in time. When this occurs, the node must give priority to one pod over another. The facility used to make this decision is referred to as a Quality of Service (QoS) Class.

7.3.1 Configure quality of service for pods

The Quality of service (QoS) class in Red Hat OpenShift can be modified by changing the Resources stanza in the deployment YAML of the workload. In an overcommitted environment when pods on one node try and use more resources than available at any given point in time, OpenShift allocates the resources based on the QoS class defined for each pod. This OpenShift document describes how these resource settings determine the quality of service (QoS) – essentially the CPU and memory allocation algorithm that is used to determine the amount of physical CPU power and memory that needs to be allocated to the pod.

OpenShift has established the following three tiers of service:

- Guaranteed QoS
- Burstable QoS
- Best Effort QoS

Table 7-1 summarizes the settings for each tier defined in the QOS settings.

Table 7-1 QOS tiers

QoS Name	Priority	Resource stanza parameters	Description / Use
Guaranteed	1-highest	limits = requests	High priority, time sensitive tasks
Burstable	2	limits > requests	Most common workloads – maximize instantaneous access to vCPU access
BestEffort	3-Lowest	Not set	Low priority tasks – first to be terminated if the system runs out of resources - background system housekeeping

The QOS setting can have a significant impact on the performance of your POD. In a recent study[2], a team was surprised by the significance of the effect of changing the Resources stanza in the deployment YAML of the workload. In the study, the team found that switching from BestEffort QOS to Burstable QOS. OpenShift Container Platform achieved over two times the throughput for a POD.

7.4 Management

Managing modern infrastructures on IBM Power systems requires a strategic approach that leverages advanced technologies and best practices. One key strategy is cloud integration, which involves migrating workloads to hybrid or multi-cloud environments. This allows organizations to benefit from the scalability, flexibility, and cost-efficiency of cloud services while maintaining the robust performance and security of IBM Power systems. Additionally, automation plays a crucial role in streamlining operations, reducing manual intervention, and enhancing overall efficiency. By automating routine tasks such as system updates, backups, and performance monitoring, IT teams can focus on higher-value projects that drive business growth.

Another important strategy is data analytics and optimization. IBM Power systems are designed to handle large volumes of data, making them ideal for running complex analytics workloads. By implementing AI-driven analytics tools, organizations can gain deeper insights into their operations, optimize resource allocation, and improve decision-making processes. These tools can analyze historical data to identify trends, predict future demands, and recommend actions to enhance performance and efficiency. Furthermore, security and compliance are critical aspects of managing modern infrastructures. IBM Power systems offer robust security features, but integrating AI-driven security solutions can provide real-time threat detection and response, ensuring that systems remain secure and compliant with industry regulations.

[2] Containers, Kubernetes, OpenShift on Power, https://community.ibm.com/community/user/powerdeveloper/blogs/mithun-h-r/2024/12/13/run-mongodb-and-nodejs-on-red-hat-openshift

The complexities of modern infrastructure necessitate the use of AI-driven tools to keep up with evolving demands.

- IBM Watson AIOps is a powerful solution that combines event management, incident diagnosis, incident resolution, and insight delivery into a single platform. It uses AI to predict, communicate, and resolve events before they become serious problems, enhancing system reliability and reducing downtime.
- Red Hat Advanced Cluster Management (RHACM) for Kubernetes provides centralized management of clusters and applications with built-in security policies. It enables organizations to deploy apps, manage multiple clusters, and enforce policies across diverse environments, ensuring consistency and compliance.

By leveraging these AI-driven tools, organizations can manage their IBM Power infrastructures more effectively, ensuring they remain agile, efficient, and resilient in the face of growing complexities.

7.4.1 IBM Watson AIOps

IBM Watson AIOps is an AI-powered platform designed to automate IT operations, predict and prevent outages, and optimize resource utilization across complex hybrid cloud environments. It leverages artificial intelligence and machine learning to analyze vast amounts of operational data, identify patterns, and provide actionable insights to IT teams. The core aim of Watsonx AIOps is to move from reactive incident management to proactive problem resolution, ultimately improving application availability, performance, and operational efficiency while reducing costs.

At its heart, Watson AIOps ingests and correlates data from various IT sources, including monitoring tools (like IBM Instana and Prometheus), logging systems, event management platforms, service management tools (like ServiceNow), and change management systems. By applying advanced AI algorithms, it can detect anomalies, identify root causes of issues, predict potential incidents before they occur, and recommend optimal remediation steps. The platform builds a dynamic and holistic view of the IT environment, understanding the relationships between applications, infrastructure, and services. This contextual awareness is crucial for effective problem diagnosis and resolution in today's increasingly complex and interconnected IT landscapes.

IBM Watson AIOps combines a set of capabilities to provide a single solution that facilitates predicting, communicating, and resolving events before they become serious problems. Issues are inevitable in any IT landscape, it's how, and when, you respond that can make a world of difference to your organization.

The core functions of 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 capabilities that manage all facets of the AIOps lifecycle from model training to execution. Figure 7-15 on page 217 illustrates how these capabilities map to Watson AIOps.

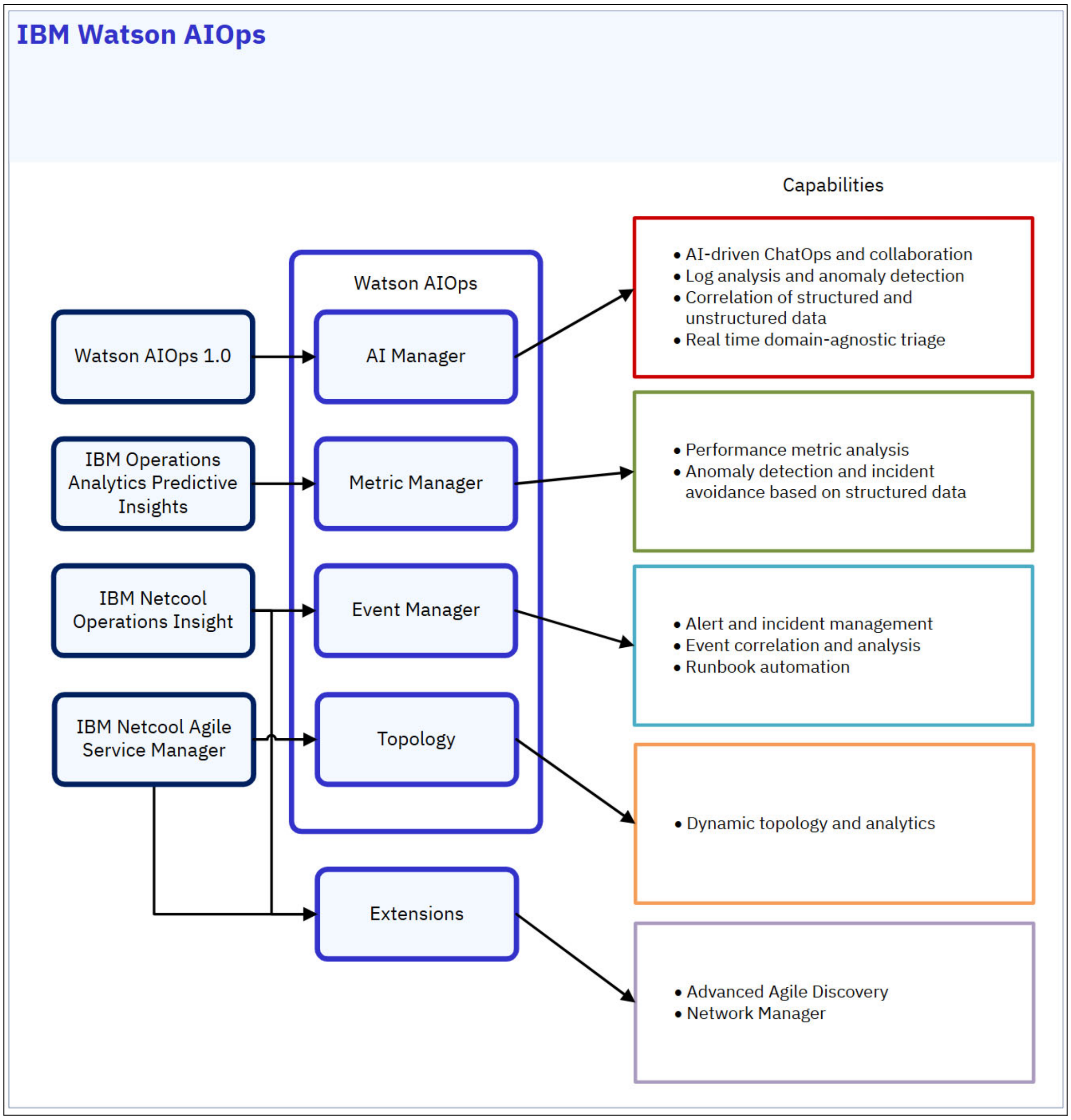

IBM Watson AIOps
Capabilities
Watson AIOps
Watson AIOps 1.0
AI Manager
IBM Operations Analytics Predictive Insights
Metric Manager
IBM Netcool Operations Insight
Event Manager
IBM Netcool Agile Service Manager
Topology
Extensions
• AI-driven ChatOps and collaboration
• Log analysis and anomaly detection
• Correlation of structured and unstructured data
• Real time domain-agnostic triage
• Performance metric analysis
• Anomaly detection and incident avoidance based on structured data
• Alert and incident management
• Event correlation and analysis
• Runbook automation
• Dynamic topology and analytics
• Advanced Agile Discovery
• Network Manager

Figure 7-15 Components of watsonx AIOps

Note: IBM Watson AIOps is available as an IBM Cloud Pak. For details and documentation, see IBM Cloud Pak for Watson AIOps.

Deriving insight into IT operations with Watson AIOps AI Manager

n today's rapidly evolving technological landscape, Chief Information Officers face a persistent challenge: how to foster innovation while maintaining a stable and reliable IT environment. This balancing act is further complicated by increasing complexity, the need to scale operations effectively, and the ongoing demand for specialized skills to keep pace with the ever-changing IT landscape. IBM Watsonx AIOps AI Manager directly addresses these critical pain points, empowering CIOs and Site Reliability Engineers (SREs) to navigate this

By leveraging the power of artificial intelligence, AI Manager intelligently analyzes vast quantities of operational data from diverse sources, including system logs, performance metrics, and event streams, to uncover hidden patterns and critical insights. This deep analysis transcends traditional monitoring approaches, revealing underlying issues and potential risks that might otherwise remain unnoticed. Importantly, these actionable insights are not confined to a separate platform but are seamlessly delivered in near real-time directly into the collaboration and workflow tools that IT teams already utilize, such as Slack. This integration ensures that the right information reaches the right people at the right time, facilitating faster awareness and more effective collaboration.

The result is an unprecedented level of visibility into an organization's entire IT infrastructure, providing a holistic understanding of its health and interdependencies. This enhanced visibility empowers IT teams to proactively identify potential failures before they impact critical services, shifting from a reactive posture to a predictive one. Furthermore, when issues do arise, AI Manager accelerates the process of problem resolution by providing intelligent root cause analysis and actionable recommendations, ultimately minimizing downtime, improving application availability, and enabling CIOs to confidently pursue innovation on a stable and resilient foundation.

Table 7-2 shows the operational information available from Watson AIOps.

Table 7-2 Operational information available from 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 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

Users, such as administrators and site reliability engineers, face challenges as they work across a range of environments, including multiple data centers, private clouds, and public clouds that run Kubernetes clusters. Red Hat Advanced Cluster Management for Kubernetes provides the tools and capabilities 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.

Clusters and applications are all visible and managed from a single console, with built-in security policies. Run your operations from anywhere that Red Hat OpenShift runs, and manage any Kubernetes cluster in your Red Hat Advanced Cluster Management for Kubernetes:

- Offers end-to-end management, visibility, and control of cluster and application life cycle, along with improved security and compliance of the entire Kubernetes domain – across multiple datacenters and public cloud environments.
- Provides a hybrid cloud management platform and capabilities that address common challenges faced by administrators and site reliability engineers (SREs) as they work across a range of environments such as multiple datacenters and private and public cloud environments that run Kubernetes clusters – including remote edge sites.
- Provides FIPS mode support and allows management of Kubernetes clusters from one place. Kubernetes clusters can range from public cloud (AWS, Google, Microsoft), private cloud (OpenStack, Virtualization) to on-premises (bare metal servers with x86_64, IBM Power, and LinuxONE or Z systems) and even to edge environments (ARM).
- Multi-cluster observability for cluster health along with optimization capabilities deliver an enhanced SRE experience with out-of-the-box multi cluster dashboards that can store long-term historical data and provide an overview of your full cloud-ready.
- Unified multi cluster life cycle management allows the creation, upgrade, and destruction of Kubernetes clusters reliably, consistently, and at scale, using an open-source programming model that supports and encourages infrastructure as code (IaC) best practices and design principles.
- Policy-based governance, risk, and compliance allows the application of a policy-based governance approach to automatically monitor and ensure desired best practices configuration state for controls related to security, resiliency, and software engineering so that these controls are operated to industry compliance standards or self-imposed corporate standards.
- Advanced application lifecycle management integrates open standards and deploys applications using placement rules that are integrated into existing CI/CD pipelines and governance controls.
- Allows Edge management at scale. With single-node Red Hat OpenShift clusters and Red Hat Advanced Cluster Management, continuously scale while enabling availability in high-latency, low-bandwidth edge use cases.
- Provides Business Continuity. Using Red Hat Advanced Cluster Management along with the broader Red Hat portfolio ensures that the applications and stateful applications your business relies on are always up and running.

Architecture

Red Hat Advanced Cluster Management for Kubernetes consists of several multi-cluster components, which are used to access and manage your clusters. The hub cluster is the common term that is used to define the central controller that runs in a Red Hat Advanced Cluster Management for Kubernetes cluster. From the hub cluster, you can access the console and product components, as well as APIs such as the rcm-api, which handles API requests related to cluster lifecycle management.

Figure 7-16 shows the hub cluster and a single managed cluster. RHACM can manage multiple clusters from a single hub cluster.

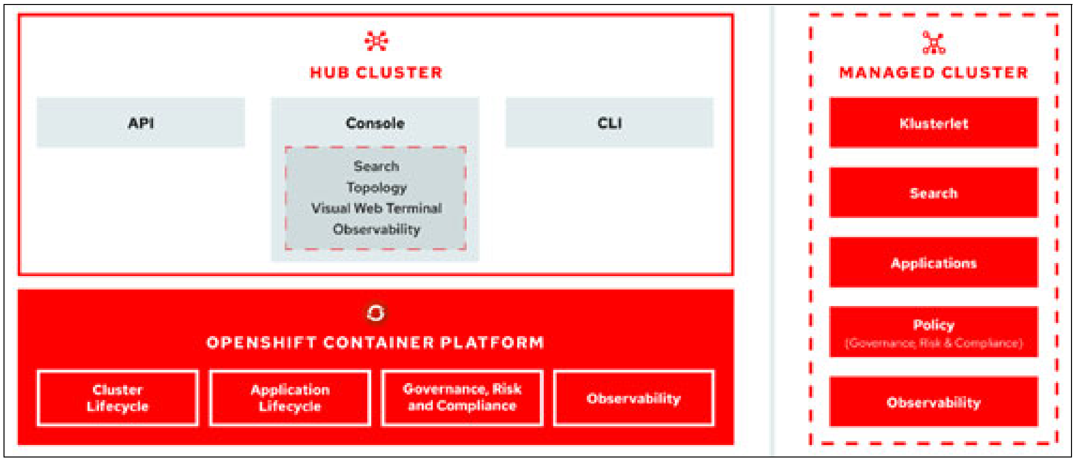


Figure 7-16 RHACM hub cluster and managed cluster[3]

The hub cluster aggregates information from multiple clusters and maintains the state of the managed clusters and the applications that run them. RHACM provides a set of REST APIs to support the various functions it uses for management. A managed cluster is one of the clusters which are managed by the hub cluster. The *Klusterlet* agent manages the connection to the hub cluster.

Figure 7-17[4] shows a more detailed view of the HRHACM components.

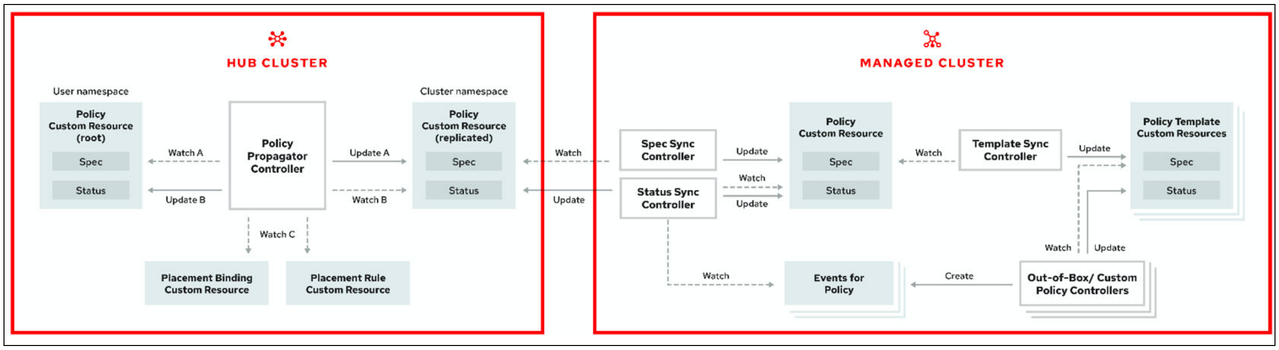


Figure 7-17 Cluster components

The RHACM hub cluster is the central controller providing the console and product components. The hub cluster uses APIs to manages API requests related to cluster lifecycle management. The APIs can look for resources across the clusters, run commands from the Visual Web Terminal, and view the topology of your environment. The hub cluster provides observability and collects metrics from your managed clusters and cloud providers.

[3] 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] https://docs.redhat.com/en/documentation/red_hat_advanced_cluster_management_for_kubernetes/2.2/html/security/governance-and-risk

The hub cluster aggregates information from multiple clusters using an asynchronous work request model. The hub cluster maintains the state of clusters and the applications running on those clusters using a graph database and it uses etcd, a distributed key value store, to store the state of work requests and results from multiple clusters.

The managed cluster receives and applies requests, then returns the results. It sends metrics to the hub cluster using the observability service.

RHACM can create clusters and import existing clusters as well as upgrade, destroy, and manage clusters across on-premise, public and private clouds. providing an aggregated view of all cluster health statuses or a view of individual health metrics of any managed cluster.

RHACM helps manage application lifecycle and can manage application resources on managed clusters. This includes automation of the deployment and lifecycle management of resources across the managed clusters. It exposes easy-to-reconcile service routes and provides full control of Kubernetes resource updates to manage all aspects of the application.

RHACM allows governance and risk management by defining the processes that are used to manage security and compliance from a central interface page. Cluster managers can view and create policies with the Red Hat Advanced Cluster Management policy framework, create RBAC controls, and define security policies.

The observability component is a service used to understand the health and utilization of clusters across a fleet. By default, multicluster observability operator is enabled during the installation of Red Hat Advanced Cluster Management. Observability on RHACM includes the definition for multiple components:

- Observability architecture
- Observability configuration
- Enabling the observability service
- Using observability
- Customizing observability
- Observability alerts
- Searching in the console introduction
- Using observability with Red Hat Insights

Observability architecture

You can use Red Hat Advanced Cluster Management for Kubernetes to gain insight and optimize your managed clusters. Enable the observability service operator, multicluster-observability-operator, to monitor the health of your managed clusters. By default, observability is included with the product installation, but not enabled due to the requirement for persistent storage.

The multiclusterhub-operator enables the multicluster-observability-operator pod by default. User must configure the multicluster-observability-operator pod. When the Observability service is enabled by defining a MultiClusterObservability custom resource, the observability-endpoint-operator is automatically deployed to each imported or created managed cluster.

This controller starts a metrics collector that collects the data from Red Hat OpenShift Container Platform *Prometheus*, then sends the data to the Red Hat Advanced Cluster Management hub cluster.

Figure 7-18 illustrates the observability components

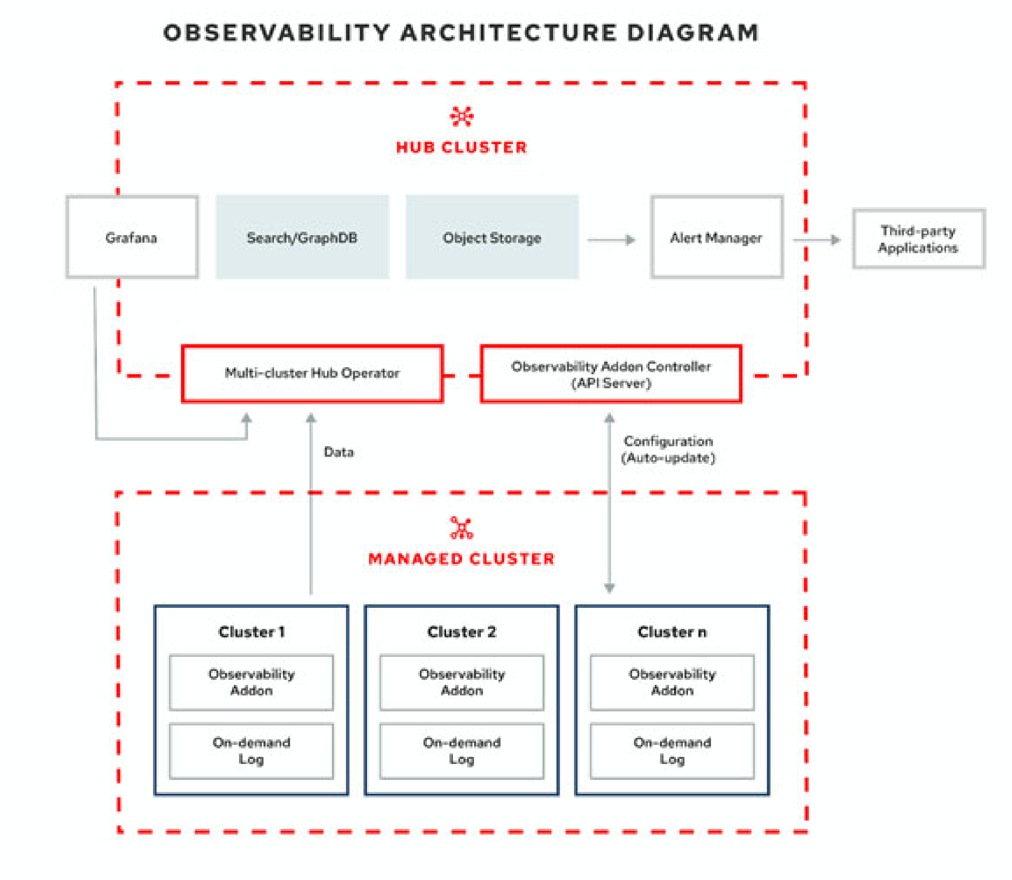


Figure 7-18 Observability Architecture diagram[5]

When the Observability service is enabled, the hub cluster is always configured to collect and send metrics to the configured Thanos instance, regardless of whether hub self-management is enabled or not. When the hub cluster is self-managed, the disableHubSelfManagement parameter is set to false, which is the default setting. Metrics and alerts for the hub cluster appear in the local-cluster namespace. The local-cluster only appears in the cluster list drop-down menu if hub self-management is enabled. You can query the local-cluster metrics in 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 your managed clusters.
- The observability add-on controller is the API server that 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 ensures that queries are performing well by using the retention configuration, and compaction of the data in storage. To help identify when the Thanos Compactor is experiencing issues, use the four default alerts that are monitoring its health.

[5] https://docs.redhat.com/en/documentation/red_hat_advanced_cluster_management_for_kubernetes/2.3/html-single/observability/index#observing-environments-intro

The observability component deploys an instance of Grafana to enable data visualization with dashboards (static) or data exploration. Red Hat Advanced Cluster Management supports version 8.5.20 of Grafana. You can also design your Grafana dashboard. For more information, see Designing your Grafana dashboard.

Alert Manager

The Prometheus Alertmanager enables alerts to be forwarded with third-party applications. You can customize the observability service by creating custom recording rules or alerting rules. Red Hat Advanced Cluster Management supports version 0.25 of Prometheus Alertmanager. Figure 7-19 illustrates how the alert manager functions.

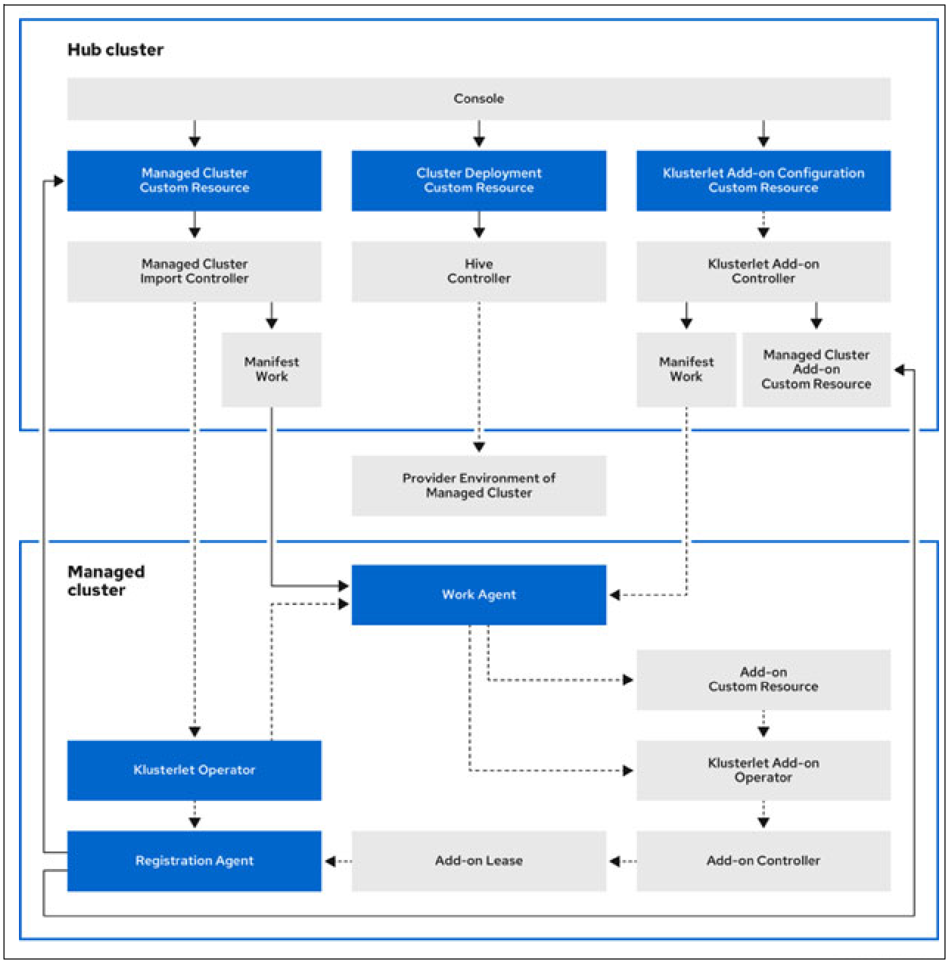

Hub cluster
Console
Managed Cluster Custom Resource
Cluster Deployment Custom Resource
Klusterlet Add-on Configuration Custom Resource
Managed Cluster Import Controller
Hive Controller
Klusterlet Add-on Controller
Manifest Work
Manifest Work
Managed Cluster Add-on Custom Resource
Provider Environment of Managed Cluster
Managed cluster
Work Agent
Add-on Custom Resource
Klusterlet Operator
Klusterlet Add-on Operator
Registration Agent
Add-on Lease
Add-on Controller

Figure 7-19 High level alert manager[6]

There are predefined alerts that are managed. Consult Table 7-3 for a list of default alerts.

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 percent.

[6] https://docs.redhat.com/en/documentation/red_hat_advanced_cluster_management_for_kubernetes/2.5/html/clusters/managing-your-clusters

Alert	Severity	Description
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 last 24 hours.

Sizing for Red Hat Advanced Cluster Management

The sizing requirements for Red Hat Advanced Cluster Manager is shown in Table 7-4. These are the suggested minimum compute resources in addition to any Red Hat OpenShift Container Platform requirement:

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 Red Hat OpenShift Container Platform sizing guidelines	Per Red Hat OpenShift sizing guidelines
Worker	3	redisgraph redis x 1	16 GB	6 CPU

Part 2

Modernization Support By Operating System

Modernization occurs across several layers within the Power infrastructure. We've explored how IBM has integrated modern tools and capabilities into the Power hardware, discussed the modernization features within the management layer, and reviewed the technologies and frameworks to help modernize your applications. Now, let’s examine how IBM has enhanced the operating systems running on IBM Power, specifically AIX, IBM i, and Linux.

These operating systems continue to evolve by incorporating advancements in key areas such as security, encryption, automation, and support for modern application development tools. These improvements are designed to facilitate the transition to a modernized infrastructure for our clients.

The following chapters are in this part.

- Chapter 8, “AIX” on page 227
- Chapter 9, “IBM i” on page 265
- Chapter 10, “Linux and OpenShift” on page 325

© Copyright IBM Corp. 2025.

8

AIX

IBM AIX is a robust and dependable operating system designed for mission-critical applications in enterprise settings. With over three decades of proven reliability, AIX provides a stable, secure, and scalable platform that supports businesses across various industries.

AIX on Power enables innovation through hybrid-cloud and open-source capabilities, offering businesses 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, incorporating new hybrid multi-cloud features and enhanced open-source capabilities.

Customers using IBM AIX benefit from improved workload scalability, better cloud automation with Ansible, enhanced security, flexible licensing options, and access to over 300 open-source packages. IBM Power Systems remains dedicated to delivering ongoing improvements to AIX, focusing on performance, scalability, resilience, and continuous innovation.

The following topics are covered in this chapter:

© Copyright IBM Corp. 2025.

8.1 Introduction to AIX

IBM AIX (Advanced Interactive eXecutive) is a proprietary UNIX operating system developed by IBM, specifically designed for IBM Power servers. First introduced in 1986, AIX, based on UNIX System V, has continually evolved, with the latest version, AIX 7.3, providing a stable and reliable platform for mission-critical workloads over the past three decades.

IBM Power has remained at the forefront of innovation, prioritizing performance, resilience, scalability, and security, solidifying its reputation as a leading server platform. AIX ensures investment protection for customers through binary compatibility and long release lifecycles. Additionally, it supports the adoption of modern technologies with flexible, subscription-based models.

In response to the growing sophistication of cyber threats, AIX 7.3 features advanced security enhancements to protect data. IBM PowerSC further bolsters security by addressing complex threats, misconfiguration, and simplifying administrative tasks, while also streamlining compliance efforts.

As IBM Power continues to expand its capabilities in hybrid cloud, AI, and cloud-native applications, AIX remains a critical component of the strategy. With a roadmap and support plan extending beyond 2035 IBM demonstrates its commitment to ensuring the long-term viability of AIX. The current roadmap is shown in Figure 8-1.

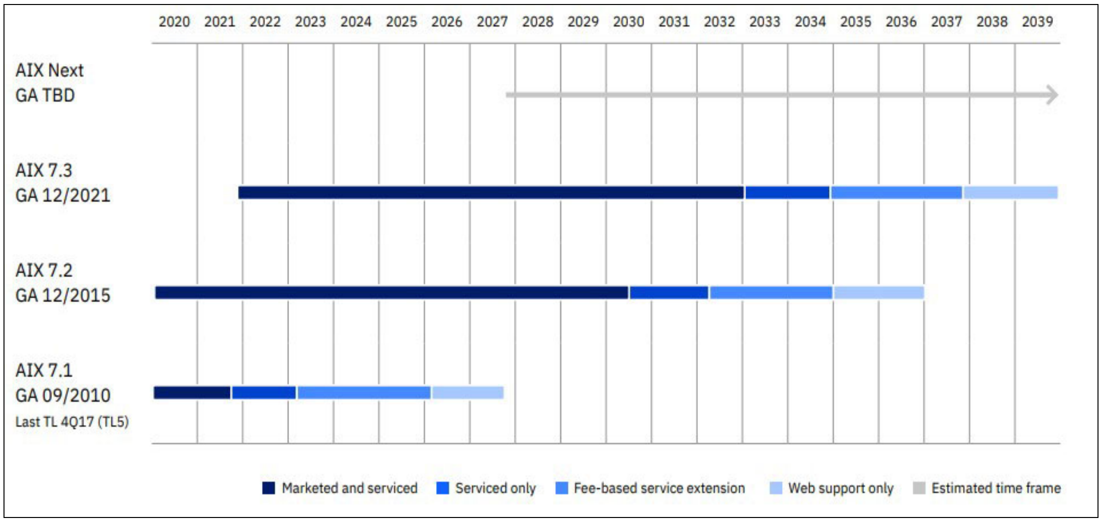


Figure 8-1 AIX roadmap

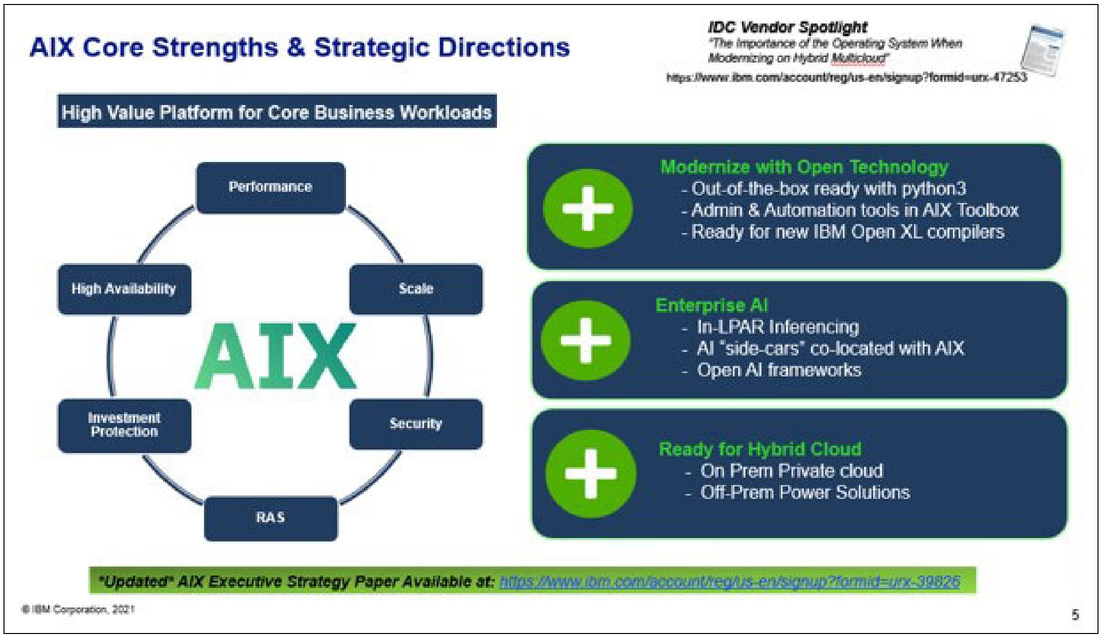
For more information on IBM's strategy and roadmap for IBM AIX see An executive guide to the strategy and roadmap for the IBM AIX operating system for IBM Power servers.

8.2 AIX Advantages

IBM is committed to supporting the thousands of businesses that depend on AIX for their most mission-critical applications and databases. This dedication 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 by users for its consistent delivery of top-tier performance, scalability, availability, and security for their most essential workloads.

With a roadmap and ongoing support extending beyond a decade, IBM affirms its commitment to innovation, particularly in hybrid cloud, AI, and other emerging technologies, ensuring AIX continues to meet the evolving needs of its users and their businesses.

AIX core strengths and strategic directions are summarized in Figure 8-2.



AIX Core Strengths & Strategic Directions

IDC Vendor Spotlight
"The Importance of the Operating System When Modernizing on Hybrid Multicloud"
https://www.ibm.com/account/reg/us-en/signup?formid=urx-47253

High Value Platform for Core Business Workloads

AIX

Performance

Scale

Security

RAS

Investment Protection

High Availability

Modernize with Open Technology
- Out-of-the-box ready with python3
- Admin & Automation tools in AIX Toolbox
- Ready for new IBM Open XL compilers

Enterprise AI
- In-LPAR Inferencing
- AI "side-cars" co-located with AIX
- Open AI frameworks

Ready for Hybrid Cloud
- On Prem Private cloud
- Off-Prem Power Solutions

"Updated" AIX Executive Strategy Paper Available at: https://www.ibm.com/account/reg/us-en/signup?formid=urx-39826

© IBM Corporation, 2021

5

Figure 8-2 AIX Core Strengths & Strategic Directions

8.2.1 RAS

In computing, RAS stands for Reliability, Availability, and Serviceability. These are key attributes of a computer system, especially for enterprise and mission-critical environments. Here's a breakdown of each component:

- Reliability: Refers to the system's ability to perform its intended functions without failure over a specified period. A reliable system minimizes the chances of errors and hardware failures, ensuring consistent performance.
- Availability: Refers to the ability of the system to be up and running, accessible, and ready for use when needed. High availability ensures that services or applications are available to users with minimal downtime, often through redundancy, fault tolerance, or failover mechanisms.
- Serviceability: Refers to the ease with which the system can be maintained and repaired. Serviceable systems are designed for easy diagnostics, repairs, and upgrades, minimizing downtime and improving the overall lifespan of the hardware.

RAS is a crucial set of features for systems used in industries where downtime can be costly or dangerous, such as in data centers, telecommunications, and enterprise applications. Systems designed with high RAS characteristics aim to provide uninterrupted service, recover quickly from failures, and ensure ongoing operations with minimal effort from administrators.

For AIX running on IBM Power, RAS is a core focus, ensuring these servers deliver the high standards of performance, uptime, and maintainability that enterprise and mission-critical workloads demand. IBM Power Systems, widely used in industries such as banking, telecommunications, and health care, are designed with robust RAS features to maximize system efficiency and minimize downtime.

Reliability

Reliability refers to how consistently the system performs its tasks without encountering failures. AIX running on IBM Power Systems is built with hardware and software features that work together to ensure a high degree of reliability. Some key elements that contribute to reliability include:

- Advanced hardware design

  IBM Power servers feature advanced error detection and correction capabilities at the hardware level. For example, memory mirroring and ECC (Error-Correcting Code) memory help protect against data corruption by automatically detecting and correcting errors in memory.

- Fault-tolerant components

  IBM Power Systems are designed with redundant components such as power supplies and cooling units to prevent single points of failure. In addition, Power10 processors are designed with capabilities to detect and mitigate errors at the chip level, contributing to reduced application downtime. This allows the system to continue operating smoothly even if individual components fail.

- AIX reliability

  The AIX operating system provides software reliability with infrequent failures due to extensive recovery capabilities – self healing. When a problem occurs, AIX does not just fail or stop running, but instead attempts to keep running, self-heals the process, alerts and logs information explaining the condition. Features such as the System Resource Controller, Logical Volume manager, Error Reporting and reliable scalable cluster technology are all built into AIX to help achieve this objective.

Availability

Availability focuses on ensuring the system is continuously operational and accessible when needed. IBM Power Systems are engineered for high availability (HA), which is crucial in industries requiring continuous operations. IBM incorporates several features to maximize availability:

- Live Partition Mobility (LPM)

  This feature allows workloads to be moved between servers without any downtime. It supports load balancing, proactive maintenance, and disaster recovery.

- PowerHA

  IBM's PowerHA software provides high availability and disaster recovery solutions. PowerHA ensures that applications and data remain accessible in case of hardware or software failures by enabling clustering and automated failover.

- Automatic Restart and Self-Healing

  IBM Power Systems are equipped with automatic restart features, which can detect faults and automatically restart components to restore functionality. The system can self-diagnose and correct many issues without requiring human intervention, ensuring that the environment remains available.

- Live kernel update

  An outstanding feature of AIX is the dynamic UNIX kernel. Unlike other UNIX variants, IBM AIX is able to accept kernel changes without compilation or reboot. The kernel is dynamically extendable and can be expanded by adding routines that belong to any of the functional classes. A process executing in user mode can customize the kernel by using the sysconfig subroutine, if the process has appropriate privilege. A user-mode process can then load, unload, initialize, or terminate kernel routines. Kernel configuration can also be altered by changing tunable system parameters.

These kernel extensions can be additional kernel services, system calls, device drivers, or File systems in Operating system and device management. The sysconfig subroutine also provides the ability to read and set system run-time operating parameters.

> **Note:** for more information on the abilities of the AIX kernel please visit these reference sites.
>
> https://www.ibm.com/docs/en/aix/7.3?topic=concepts-kernel-environment
> https://www.ibm.com/docs/en/aix/7.3?topic=s-sysconfig-subroutine
> https://www.ibm.com/docs/en/aix/7.3?topic=aix-kernel-extensions-device-support-programming-concepts

Serviceability

Serviceability is about how easily a system can be repaired, maintained, and upgraded, minimizing downtime for administrators and ensuring the system’s longevity. IBM Power systems are designed with serviceability in mind through various features:

- Predictive Failure Analysis (PFA)

  IBM Power servers include predictive failure analysis that can detect and report issues before they result in a failure. This enables system administrators to replace parts proactively, reducing unplanned downtime and preventing service interruptions. The system

- First Failure Data Capture

  First Failure Data Capture (FFDC) is a mechanism to capture system information when a severe, unrecoverable error occurs, aiding in troubleshooting and diagnosis. The primary goal of FFDC is to provide IBM support teams with the necessary data to understand and resolve critical issues quickly and effectively. FFDC logs crucial system information when a severe error occurs that prevents AIX from functioning correctly.

- Remote diagnostics and management

  IBM Power servers come with a suite of remote management tools, such as IBM Systems Director and IBM IMM (Integrated Management Module), which enable administrators to monitor the system’s health, identify issues, and manage hardware and software remotely. This improves system uptime and reduces the time required for on-site service.

- Hot-Swappable Components

  Many IBM Power servers allow administrators to replace components such as power supplies, fans, and hard drives without having to shut down the system. This makes repairs and upgrades faster and easier without affecting system availability.

8.2.2 Scalability and dynamic management

AIX excels at dynamic adaptability. It seamlessly integrates new devices and accommodates configuration changes in real-time, minimizing disruptions and maintaining peak performance. For enhanced uptime and scalability, AIX offers robust high-availability clustering, allowing you to distribute workloads and ensure continuous operation

RSCT/RMC

The RSCT/RMC implementation provides a high availability infrastructure for managing resources in a standalone system, as well as in a cluster (peer domain or management domain). Reliable Scalable Cluster Technology (RSCT) is a set of software components that together provide a comprehensive clustering environment for AIX and Linux. RSCT is used by a variety of IBM products to provide cluster infrastructures with improved system availability, scalability, and ease of use. RSCT includes the following components:

- Resource Monitoring and Control (ctrmc) subsystem
- RSCT core resource managers (ctcas)
- SCT cluster security services (ctsec)
- Topology Services subsystem (cthatsd)
- Group Services subsystem (cthagsd)

Note: For more information on RSCT concepts refer to
`https://www.ibm.com/docs/en/rsct/3.3?topic=administering-concepts`

Object Data Manager

The Object Data Manager (ODM) is another unique feature of the AIX operating system. ODM is a data manager intended for storing system information. Information is stored and maintained as objects with associated characteristics. This mechanism ensures the integrity of the AIX operating system, and does not allow system administrators to configure unsupported attributes of devices and the AIX operating system. ODM is customizable and adds to the extensibility of AIX.

Other UNIX systems use ASCII stanza files that can be manually edited when configuring a driver into the kernel. In contrast, AIX also utilizes stanza files, but these are stored in a directory known as the ODM database. Unlike other systems, you do not directly modify the files in the ODM database in AIX. Instead, you use ODM routines and commands to make changes. Typically, you won’t need to manually use ODM commands, as standard system utilities like **cfgmg**r, **mkdev,** and **rmdev** automatically call the ODM routines to keep the ODM database updated.

A device configuration method in AIX would be invoked by a root user running the **cfgmgr** command, or called by **rc.boot** in RAM disk when the computer system is started. This device method would eventually call the sysconfig subroutine referred to earlier. The below list provides the device methods called during a **cfgmgr** task and the interaction with the AIX ODM when a device is added to the AIX operating system.

- 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 ensures that a parent device exists in the CuDv object.

- Configure method (causes device to be available)

  A configure method should perform the following steps:

  - Display LED value on system LED panel
  - Verify that a parent device is available (in ODM)
  - Verify that a device is present
  - Invoke the **busresolve** system call to get an interrupt level assigned to the device.
  - Extend the kernel by calling sysconfig.

Note: For more detailed information of 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 on the ODM concepts see these documents:

- https://www.ibm.com/docs/en/aix/7.3?topic=concepts-object-data-manager
- https://www.ibm.com/docs/en/ssw_aix_72/devicemanagement/pcie_odm_ihv.pdf

8.2.3 Advanced Virtualization

AIX is built as the flagship operating system for IBM Power where the most advanced hypervisor, PowerVM is available for mission critical business requirements.

IBM PowerVM is a powerful virtualization technology that allows IBM Power Systems servers to run multiple virtual machines (VMs) concurrently. This capability enables businesses to consolidate workloads, improve server utilization, and reduce costs. PowerVM provides a secure and scalable virtualization environment, supporting a range of operating systems, including AIX, IBM i, and Linux.

Essentially, PowerVM creates logical partitions (LPARs) within a physical server, each functioning as an independent system. This allows for flexible resource allocation, where processors, memory, and I/O can be dynamically assigned to individual VMs based on their specific needs. PowerVM also offers advanced features like Live Partition Mobility, which enables the migration of running VMs between servers without downtime, further enhancing availability and flexibility.For more information on PowerVM see the section on PowerVM.

8.3 AIX Version 7.3

AIX 7.3 brings a suite of enhancements designed to address the evolving demands of modern IT environments. Performance is a key focus, with optimizations tailored to leverage the latest Power processor capabilities, resulting in noticeable improvements in overall system speed, memory management, and I/O operations. Security is also paramount, and AIX 7.3 incorporates strengthened features to protect against emerging threats, extending security enhancements to virtualized environments. Reliability and availability are further reinforced through improvements to high-availability clustering and a general emphasis on system stability. Management is streamlined with enhanced tools and automation support, simplifying administrative tasks. Furthermore, AIX 7.3 embraces modern technologies, offering improved support for containerization and cloud environments, alongside updates to core file system functionality and size limitations. A notable enhancement is the refinement of the Live Update feature, enabling more dynamic and less disruptive changes to running logical partitions.

Here is a summary of some of the enhancements delivered in AIX 7.3:

- Extends scalability of AIX
  - Now supporting max of 240 cores
  - Increase file and filesystem capacity to 128TB for growing data needs
- Reduced Downtime
  - Reduced IPL times for large memory LPARs
  - vPMEM - allows the persistence of memory during VM restart. Avoids time-consuming memory reloads for in memory applications like SAP HANA.
  - Live kernel update improvements
    - Reduced “blackout” window
    - Overall performance improvements
    - Enable IPSEC and audit stream for LKU
    - Live LPAR profile updates
    - Console message improvements

  - Live Library Update
  - Dump time improvements
- Workload optimization
  - Integrates use of 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 (DLPAR
- Optimized security
  - More secure password policy and algorithm
  - OpenSSL performance
  - Audit subsystem performance
  - IPSEC performance
  - Extends logical volume (LV) encryption to additional devices including rootvg
  - LV/PV encryption performance
  - Opencryptoki
  - OpenSSH 9.7p1
- Administrative Efficiency
  - New automation use cases
  - AIX Toolbox updates
  - Out-of-the-box ready for Ansible with Python 3
- Streamline 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 the latest information on what is new in AIX 7.3 refer to this IBM Knowledge Center article.

8.3.1 Security updates

AIX 7.3 introduces several key security enhancements designed to bolster system integrity and protect against emerging threats. One significant improvement is the addition of more robust security controls for managing user access. This includes support for more advanced authentication mechanisms such as two-factor authentication (2FA) and tighter integration with IBM's Security Identity Governance and Intelligence (IGI) solutions. Additionally, AIX 7.3 enhances the security of system communications by introducing improvements in encryption, including better support for Transport Layer Security (TLS) protocols, ensuring that data in transit is more securely protected.

Moreover, AIX 7.3 brings enhancements to the Trusted AIX feature, which helps prevent unauthorized changes to system configurations. The operating system also includes advanced auditing capabilities that allow administrators to track system activities with finer granularity. New security policies, such as stricter user account management and better control over administrative privileges, contribute to reducing the attack surface and making systems more resistant to unauthorized access. These features, along with ongoing updates and patches from IBM, help AIX 7.3 maintain its reputation as a secure platform for enterprise computing.

Some of the IBM AIX 7.3 security enhancements are:

- New default password algorithm (SSHA-256)
- Out of the box long password support (up to 255 chars)
- Stronger default password policy
- Revised bos.net install defaults to omit packages without stronger security
- LVM Encryption for rootvg and dump devices
- RFC 7383 IKE fragmentation with IPSEC

- IPSEC support for NAT-T with IKEv2
- Trace of channel 0 now requires root privilege by default

For more information on new features in AIX 7.3 refer to What’s new in AIX security.

Trusted Execution

The removal of Trusted Computing Base (TCB) in AIX 7.2 and 7.3 signifies a shift towards a more adaptable security paradigm. TCB, while providing a foundation for system integrity, was a static feature enabled during installation, offering limited flexibility for evolving security needs. In its place, AIX now emphasizes Trusted Execution (TE), a dynamic collection of features designed to verify system integrity and enforce advanced security policies. TE offers a more granular and adaptable approach, allowing administrators to tailor security measures to specific requirements. This transition reflects a move towards a security model that can readily adapt to emerging threats and evolving compliance standards, enhancing the overall trustworthiness of the AIX environment.

A common method for a malicious user to compromise a system is by gaining unauthorized access, then installing Trojans, rootkits, or tampering with critical security files, making the system vulnerable to exploitation. The primary goal of the Trusted Execution features is to prevent these malicious activities. In the worst-case scenario, these features aim to identify if such an incident has occurred, ensuring the system remains secure and any breaches are detectable.

TE is a built in feature of AIX used to verify the integrity of the system’s executables that are allowed to run or the set of kernel extensions that are allowed to be loaded. Similar to that of Trusted Computing Base (TCB) there exists a database which is used to store critical security parameters of trusted files present on the system. This database, called Trusted Signature Database (TSD), resides in the */etc/security/tsd/tsd.dat*.

Using TE, the system administrator can then decide upon the actual set of executables that are allowed to execute or the set of kernel extensions that are allowed to be loaded. It can also be used to audit the security state of the system and identify files that have changed, thereby increasing the trusted level of the system and making it more difficult for the malicious user to do harm to the system. The set of features under TE can be grouped into the following:

- Managing Trusted Signature Database
- Auditing integrity of the Trusted Signature Database
- Configuring Security Policies
- Trusted Execution Path and Trusted Library Path

Enabling Trusted Execution

For more information on enabling Trusted Execution refer to this IBM Support document. IBM Power SC can be used to centrally manage the TE feature of multiple AIX endpoints. For more information on PowerSC and Trusted Execution refer to *Simplify Management of IT Security and Compliance with IBM PowerSC in Cloud and Virtualized Environments*, SG24-8082.

AIX Security Assessment

If your organization uses AIX or VIOS, you can be faced with the challenge of securing your systems. This AIX Security Assessment, which requires only a few hours of your time, provides a comprehensive security analysis of a single AIX or VIOS instance. This offering is designed to identify security safeguards that can be implemented to mitigate security risk on your AIX or VIOS systems.

The assessment consists of 2 components:

- Policy Assessment (optional)
- Host Assessment

For more information on requesting an AIX Security Assessment from IBM Services® refer to this website:

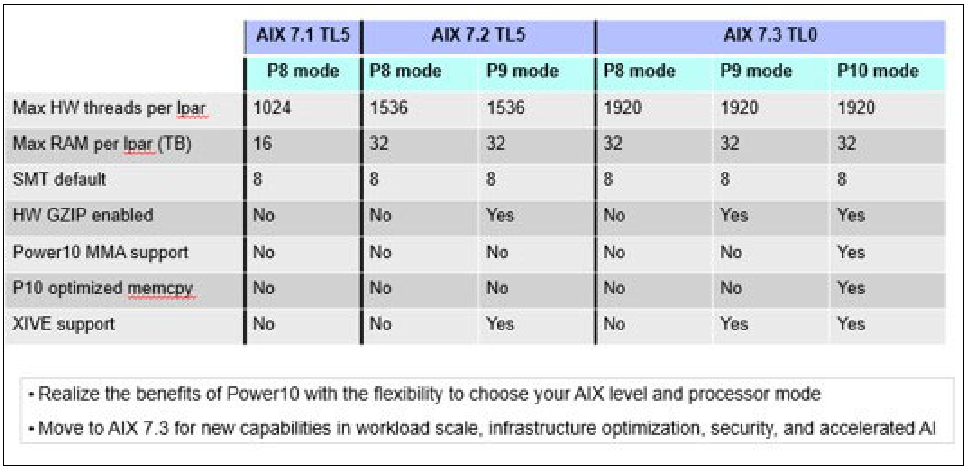
https://www.ibm.com/support/pages/aix-security-assessment

8.4 AIX Processor Compatibility Modes

IBM Power processors offer compatibility modes to ensure smooth transitions and continued support for legacy applications. These modes allow newer Power systems to execute code compiled for older architectures, minimizing the need for extensive application recompilation or migration. This backward compatibility is crucial for businesses that have invested heavily in software tailored to specific Power processor generations.

Specifically, Power processors can switch between different instruction set architectures (ISAs), effectively emulating the behavior of previous generations. This enables the execution of binaries compiled for earlier Power architectures on the latest hardware, This capability provides a bridge for customers, allowing them to upgrade their infrastructure while maintaining the functionality of their existing software investments.

Figure 8-3 displays AIX versions and capabilities 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 memcpy	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 functionality

The Power10 processor is available in both single chip module (SCM) and dual chip module (DCM) configurations.

The E1080 SCM configuration allows a maximum of 15 cores on a single socket with a maximum of 16 sockets, providing up to 240 cores. Each node has four sockets, and a maximum of four nodes are allowed in a system

The E1050 DCM configuration allows a maximum of 96 cores when using 24-core DCMs populating 4 sockets. A DCM fills one socket, and the DCMs are available with 12, 18 or 24 cores per socket. There is a maximum of one node with four sockets and multi-node configurations are not supported.

The hardware design determines the near, far, and distant NUMA memory and CPU operations, where the CPU required increases as access moves from near to far and distant hardware locations.

The AIX operating system on IBM Power systems has been optimized for general performance on IBM Power Systems through extensive testing and best practices tuning defaults set by the IBM Power Systems Performance team. These defaults have been tested against industry standard benchmarks such as SPEC, TPC, HPC and published IBM Power performance report (rPerf) performance data. Therefore, you can run AIX straight out of the box without any customization to achieve an optimal performance experience.

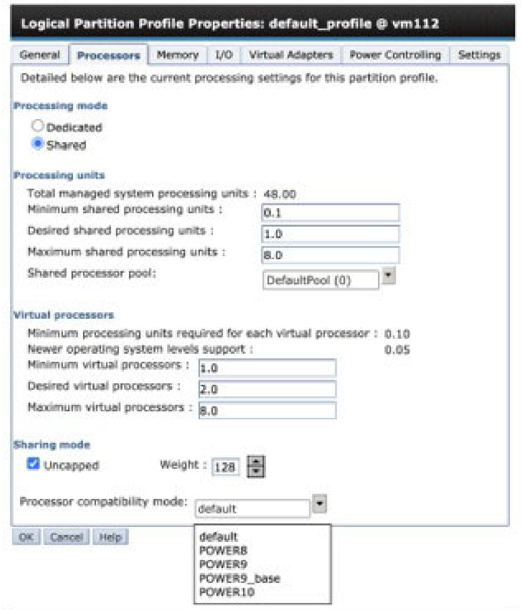
In order to avoid unexpected results where administrators still feel the need to make changes, the Power Systems Performance team has implemented the use of restricted tunables on AIX to improve the code base. These tunables can only be changed after a detailed analysis of the workload and approval by AIX Performance Development, which may result in a code change or a modification to the default tunable value

The AIX version functionality are automatically implemented by PowerVM by the processor compatibility mode determine by the AIX operating system version installed.

The hypervisor sets the effective processor compatibility mode for a logical partition by using the following information:

- The processor features supported by the operating system environment running in the logical partition.
- The configured processor compatibility mode that you specify.

Consequently, an administrator need only upgrade the AIX operating system, set the processor compatibility mode to default, and migrate to new IBM Power10 hardware using LPM to be sure of an optimal performance experience that utilizes the all the available features of the new hardware for that level of AIX, without any further operating system configuration or tuning. Figure 8-4 shows setting the processor mode in your HMC LPAR profile.



Logical Partition Profile Properties: default_profile @ vm112

Detailed below are the current processing settings for this partition profile.

Processing mode

○ Dedicated
● Shared

Processing units

Total managed system processing units : 48.00
Minimum shared processing units : 0.1
Desired shared processing units : 1.0
Maximum shared processing units : 8.0
Shared processor pool: DefaultPool (0)

Virtual processors

Minimum processing units required for each virtual processor : 0.10
Newer operating system levels support : 0.05
Minimum virtual processors : 1.0
Desired virtual processors : 2.0
Maximum virtual processors : 8.0

Sharing mode

☑ Uncapped Weight : 128

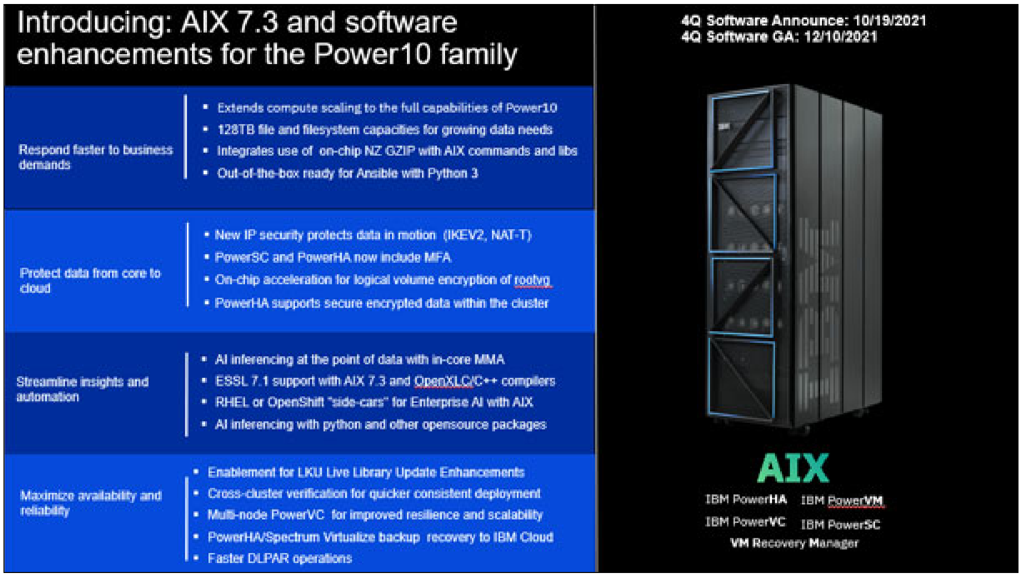
Processor compatibility mode: default

Figure 8-4 Processor modes in an HMC profile

Note: For more information on processor compatibility modes refer to this IBM document.

8.5 AIX and Power10 Support

AIX is continually enhanced to support new features provided in the IBM Power processors. AIX 7.3 provides full support for the features provided by Power10. When the next IBM Power Generation is announced, AIX will be updated to provide support for new functions provided by that generation hardware. Figure 8-5 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

Respond faster to business demands

- Extends compute scaling to the full capabilities of Power10
- 128TB file and filesystem capacities for growing data needs
- Integrates use of on-chip NZ GZIP with AIX commands and libs
- Out-of-the-box ready for Ansible with Python 3

Protect data from core to cloud

- New IP security protects data in motion (IKEV2, NAT-T)
- PowerSC and PowerHA now include MFA
- On-chip acceleration for logical volume encryption of rootvg
- PowerHA supports secure encrypted data within the cluster

Streamline 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
- RHEL or OpenShift "side-cars" for Enterprise AI with AIX
- AI inferencing with python and other opensource packages

Maximize availability and reliability

- 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 version 3.9.6. The new command to invoke python is **`/usr/bin/python3`**. The python shipped in 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.
  - JFS2 filesystem size and file size limits are increased beyond 32TB and 16TB respectively.
  - Supports the use 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 Internet Key Exchange version 2 (IKEv2)
  - IPsec support for Network Address Translation-Traversal (NAT-T) with IKEv2
  - Provides enhanced support for logical volume (LV) encryption to include encryption for logical volume support in rootvg and dump device.
  - PowerSC and PowerHA now include MFA
  - PowerHA supports encrypted data volumes within the cluster
- Streamline Insights and Automation
  - AIX 7.3 provides the capability to leverage matrix math accelerator (MMA) AI Accelerator instructions thru support of Power10 processor compatibility mode

  - IBM Open XL C/C++ for AIX is a standards-based high-performance compiler that facilitates the creation and maintenance of applications written in C and C++ for IBM Power solutions. It generates code that can take advantage of the capabilities of the latest Power 10 architecture and optimize your hardware utilization. A number of new built-in functions are delivered in this release to unlock Power 10 architecture instructions
  - Several open source packages such as pytorch and numpy, have been tested on AIX and can be used for AI use cases which enable inferencing at the point of data
  - Applications on AIX can call-out to RHEL or OpenShift “side-cars” to leverage MMA accelerated AI capability for inferencing
  - RHEL or 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 required to dynamically add processor and memory resources to a running logical partition (LPAR). --> still gathering data but expect about 30-40% improvement
  - Reduces initial program load (IPL) times for multi-terabyte memory LPARs.
  - Live kernel update supports the changing of select boot time parameters without a requirement for a reboot.
  - SMB 3.0.2 support for Live Kernel Update (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, called “AIX5L,” marked the growing integration of AIX with Linux tools and utilities, acknowledging their increasing popularity and practical value. In 2006, IBM started offering these Linux tools and utilities on separately orderable media, branding it 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 specifically built for AIX IBM Systems. It provides a robust development environment that many Linux application developers prefer. The tools are packaged using the easy-to-install RPM format. Given the strong synergy between Linux and AIX, the AIX operating system, with its long history of standards compliance, makes it relatively simple to rebuild open-source applications for AIX. This toolkit highlights the deep connection between the Linux and AIX operating systems.

8.6.1 AIX Toolbox for Open Source Software

AIX Toolbox for Open Source Software contains a collection of open source and GNU software built for AIX IBM Systems. These tools provide the basis of the development environment of choice for many Linux application developers. All the tools are packaged in RPM format.

The AIX Toolbox for OpenSource Software can be downloaded from the below URL, or an ISO image can be obtained from your Customer Entitled Support webpage.
https://www.ibm.com/servers/eserver/ess/landing/landing-page

The procedure for installing the toolbox repo and making **dnf** available on AIX is given in section “Create an AIX image in PowerVC that includes the AIX Toolbox” on page 243.

> **Note:** To download the AIX ToolBox for OpenSource Software
> https://www.ibm.com/support/pages/aix-toolbox-open-source-software-overview

bash

Let’s start with the bash shell. The bash run time environment is built into the AIX 7.3 runtime environment as shown in Example 8-1.

Example 8-1 bash shell 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 simply by running bash as shown in Example 8-2.

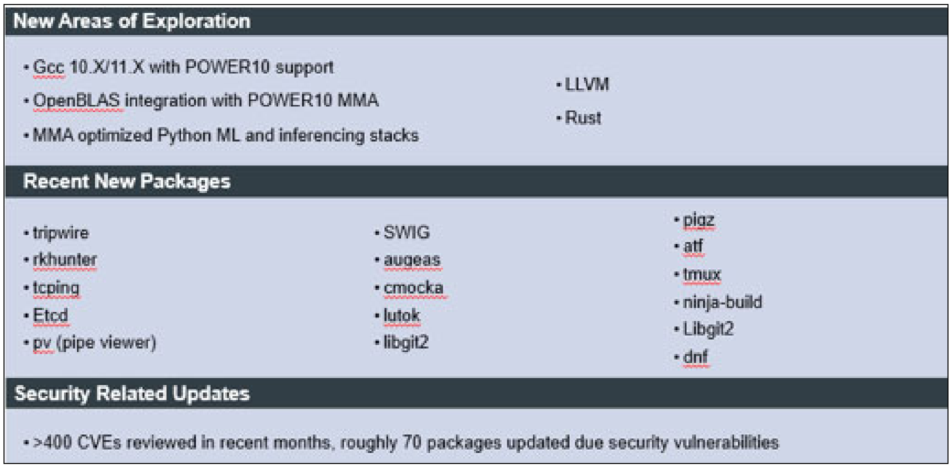
Example 8-2 Changing to bash shell

```
root@rbknim:/>bash
bash-5.2#
bash-5.2#
(reverse-i-search)`':
bash-5.2#
```

Matrix-Multiply Assist (MMA) exploitation in OpenBLAS on IBM AIX

OpenBLAS is a widely used open source BLAS library, used y the scientific community. Used to speed up linear algebra computations with low-level routines that operate on vectors and matrices with platform-specific optimizations.

Different level BLAS routines have been optimized to exploit MMA when running on Power10 processor-based systems with AIX (big endian) and Linux (little endian). OpenBLAS APIs have been internally modified to use MMA ‘C’ data types, functions, and instructions while running on Power10. To leverage the MMA benefit while running AI workloads, users have to just download the MMA-optimized OpenBLAS package from the AIX toolbox on a Power10 processor-based system and run their AI programs without any modifications. Figure 8-6 shows a list of recent additions to the OpenSource AIX and Linux Toolbox.



New Areas of Exploration

- Gcc 10.X/11.X with POWER10 support
- OpenBLAS integration with POWER10 MMA
- MMA optimized Python ML and inferencing stacks
- LLVM
- Rust

Recent New Packages

- tripwire
- rkhunter
- tcping
- Etcd
- pv (pipe viewer)
- SWIG
- augeas
- cmocka
- lutok
- 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 OpenSource AIX Linux ToolBox

The IBM AIX toolbox now provides a new **gcc** to support Power10 instructions including MMA exploitation. The provisioning of the *openblas* library with Power10 MMA integration with AIX 7.3 creates a good baseline for the collection of open source machine learning and inferencing frameworks like *pytorch, numpy,* and *scipy*, to run on AIX with Power10 acceleration.

> **Note:** For more information on the Matrix-Multiply Assist (MMA) exploitation in OpenBLAS on IBM AIX please visit OpenBlas on AIX.

Ansible Collections for AIX and IBM Power

The IBM Power Systems AIX collection provides Ansible modules to assist administrators automation of workloads and tasks on the IBM Power System. This collection can be found at `https://ibm.github.io/ansible-power-aix/.`

For additional information on the use of Ansible in the IBM Power environment, including support for AIX, refer to *Using Ansible for Automation in IBM Power Environments*, SG24-8551

There are many examples of use of this collection in periodic tasks like:

- Upgrading the VIOS, and managing filesystems as provided in the *ibm.power_vios collection* in this git repository `https://github.com/IBM/ansible-power-vios`
- The Ansible *ibm.power_hmc collection* can be used to provision or modify an LPAR through the HMC. The collection also supports other HMC commands for gathering information on managed LPARs.
- Management of AIX updates is provided as well. Using Ansible can simplify the maintenance and security updates saving your administrators time and ensuring compliance with your update strategy.

Ansible collections are easily installed on an Ansible workstation using the ansible-galaxy command. This is 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'
```

Consider a scenario where you need to automate virtual machine (VM) deployment on IBM Power Systems, but you lack an existing OpenStack environment provided by PowerVC for automated SAN storage assignment. In this case, Ansible can be leveraged to automate the crucial step of creating virtual Fibre Channel adapters. This automation generates the necessary LPAR profile and WWPNs, preparing the VM for SAN Logical Unit Number (LUN) assignment by the storage administrator. Subsequently, a boot process can recognize the SAN administrator's zoned LUNs, enabling the VM to access its storage.

Alternatively, this Ansible job can be integrated into a larger workflow, encompassing automated SAN tasks to provision the required LUNs. Essentially, Ansible bridges the gap by automating the configuration of virtual Fibre Channel adapters, facilitating 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 fiber channel adapters on each VIOS for a VM LPAR profile name `ansi_test.` The HMC password is passed using an Ansible vault

Example 8-4 Create an LPAR with virtual fiber channel 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
          61323366356163383139373862366264613032356362353336353532633031376637663463386132
          31376535316435633836303465393638636135633933364350a63333261353436636630386532366
5
          39343239373134393731626130646635663765386566613462643531323339646538303639663038
          3933366261663034300a33326262653263326164323130356538363235383038666662373064666
6
          3532

   tasks:
   - name: Create an AIX/Linux logical partition
     powervm_lpar_instance:
       hmc_host: '{{ inventory_hostname }}'
       hmc_auth: "{{ curr_hmc_auth }}"
       system_name: xxxxxxxxxx
       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: xxxxxxxxxxxxxx
        vm_name: ansi_test
        keylock: normal
        action: poweron
```

To see what is available for IBM Power and AIX, please visit the Ansible Galaxy site.

Note: For more details about the Ansible HMC modules please visit `https://galaxy.ansible.com/ui/repo/published/ibm/power_hmc/`

Note: Please visit

`https://www.ibm.com/support/pages/aix-toolbox-open-source-software-whats-new`

to find out whats new in the AIX Linux ToolBox.

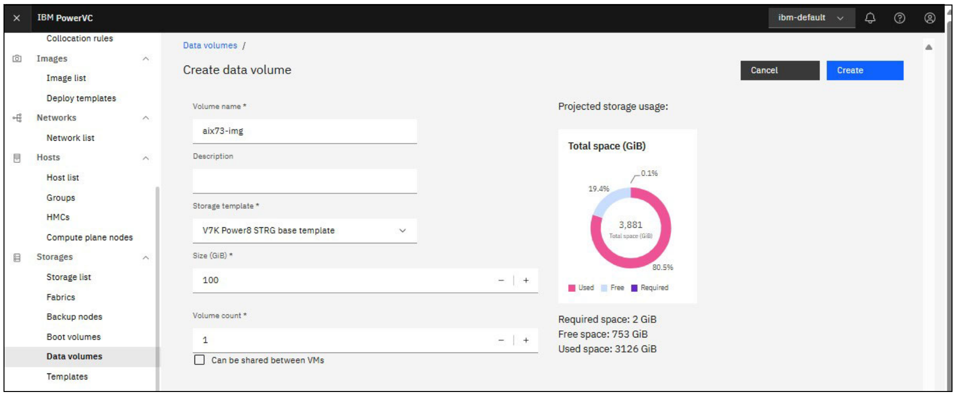
The AIX Toolbox team recommends DNF (the next-generation replacement for YUM) to install and manage Open Source software packages and dependencies from the AIX Toolbox.

Note: See this IBM blog for more information on installing DNF for the AIX OpenSource Toolbox

Create an AIX image in PowerVC that includes the AIX Toolbox

We will use IBM PowerVC to deploy a VM with the AIX OpenSource ToolBox. We start with creating an image to deploy AIX7.3.

1. Create a LUN the size of the required rootvg as shown in Figure 8-7.



Create data volume

Volume name * aix73-img

Description

Storage template * V7K Power8 STRG base template

Size (GiB) * 100

Volume count * 1

□ Can be shared between VMs

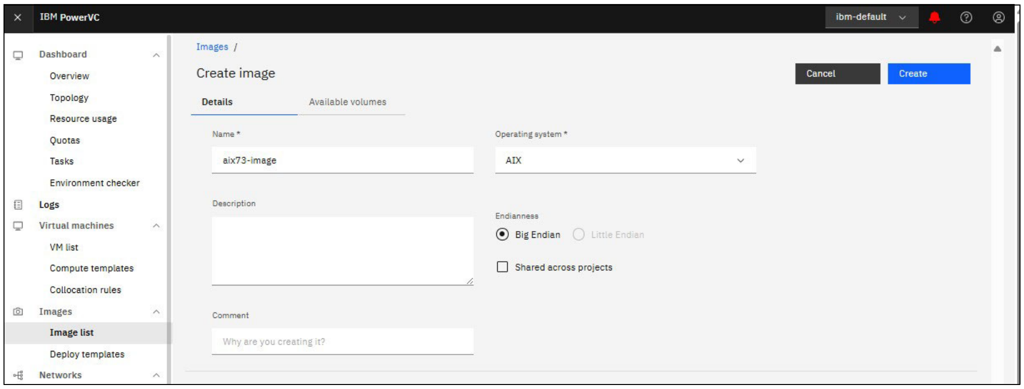
Projected storage usage:

Total space (GiB)

Required space: 2 GiB
Free space: 753 GiB
Used space: 3126 GiB

Figure 8-7 Create a LUN the size of the required rootvg

2. create a blank image for that LUN as shown in Figure 8-8.



Create image

Name * aix73-image

Operating system * AIX

Description

Endianness

● Big Endian ○ Little Endian

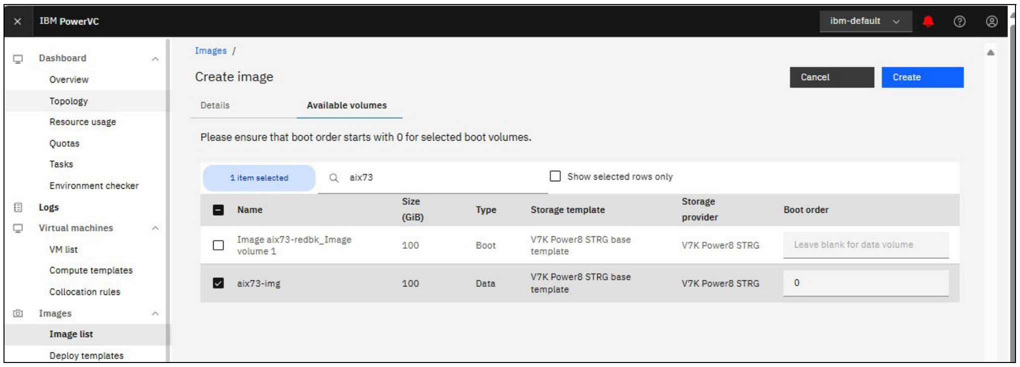
□ Shared across projects

Comment

Why are you creating it?

Figure 8-8 Create an image

3. Add the blank LUN previously created to that image. Don’t forget to add 0 to for the boot order to define it as a boot disk as shown in Figure 8-9.



Please ensure that boot order starts with 0 for selected boot volumes.

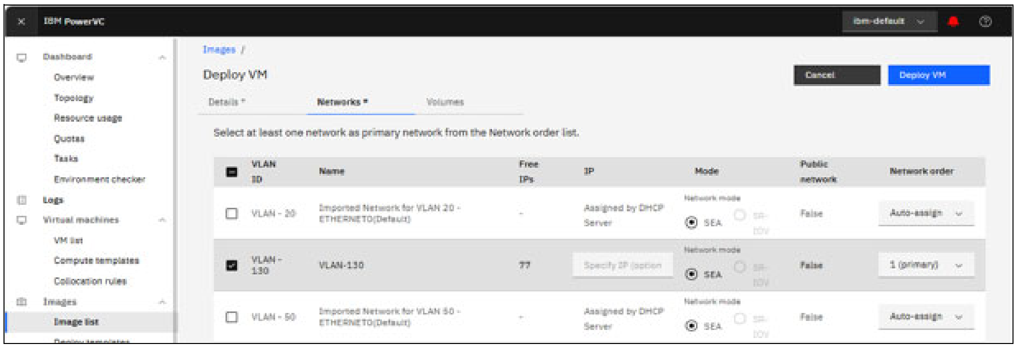
	Name	Size (GiB)	Type	Storage template	Storage provider	Boot order
☐	Image aix73-redbk_Image volume 1	100	Boot	V7K Power8 STRG base template	V7K Power8 STRG	Leave blank for data volume
☑	aix73-img	100	Data	V7K Power8 STRG base template	V7K Power8 STRG	0

Figure 8-9   Add the blank LUN to that image

4. Deploy a VM for Operating System Installation

   Now deploy a VM with that image. The VM will not have an operating system because the image is a blank disk. This step will boot to System Management Services (SMS) and allow you to attach an operating system disk for installation of the VM. At this stage you can give this VM any name, as you can chose to delete it later. We will capture this VM after it is installed, and deploy new VMs from the captured image. The original VM can be kept for change management and recapture, or just deleted.

5. Assign a network to the VM and identify the network as primary as shown in Figure 8-10.

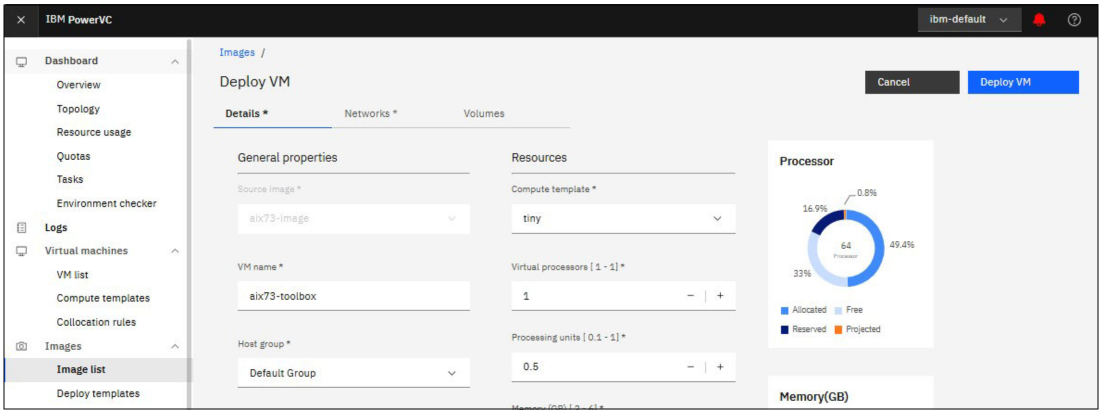


Select at least one network as primary network from the Network order list.

	VLAN ID	Name	Free IPs	IP	Mode	Public network	Network order
☐	VLAN - 20	Imported Network for VLAN 20 - ETHERNET0(Default)	-	Assigned by DHCP Server	Network mode ● SEA ○ SR-IOV	False	Auto-assign
☑	VLAN - 130	VLAN-130	77	Specify IP (option	Network mode ● SEA ○ SR-IOV	False	1 (primary)
☐	VLAN - 50	Imported Network for VLAN 50 - ETHERNET0(Default)	-	Assigned by DHCP Server	Network mode ● SEA ○ SR-IOV	False	Auto-assign

Figure 8-10   Assign a Network to the VM

6. Now deploy the blank VM as shown in Figure 8-11.



General properties

Source image *: aix73-image

VM name *: aix73-toolbox

Host group *: Default Group

Resources

Compute template *: tiny

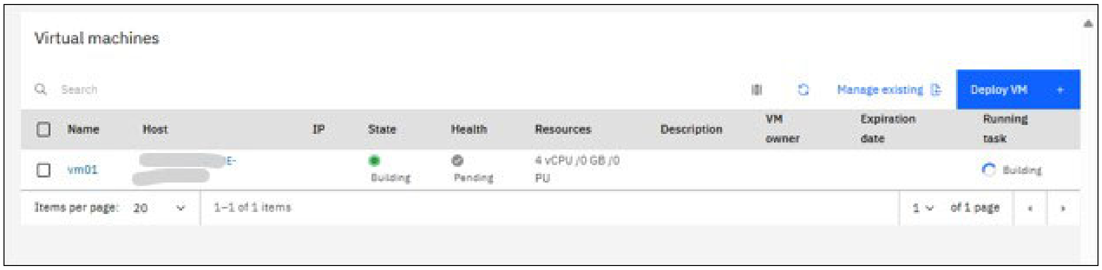
Virtual processors [1 - 1] *: 1

Processing units [0.1 - 1] *: 0.5

Memory (GB) [2 - 4] *

Figure 8-11   Deploy the blank VM

7. You will see the VM building. Wait until it gets an IP assigned to be sure it is available for login as shown in Figure 8-12.

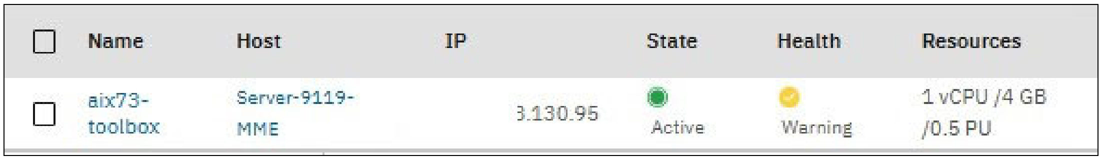


Virtual machines

☐	Name	Host	IP	State	Health	Resources	Description	VM owner	Expiration date	Running task
☐	vm01	[redacted]E-		● Building	Pending	4 vCPU /0 GB /0 PU				Building

Figure 8-12 VM Building

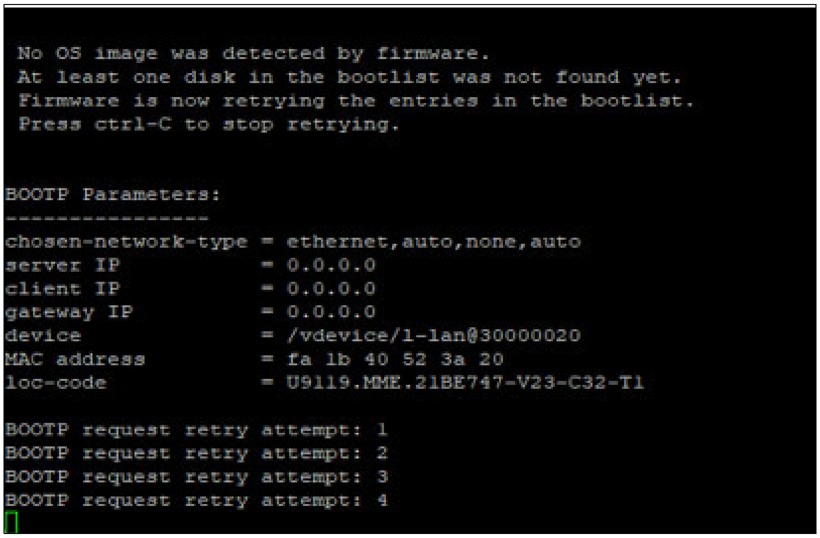
When the VM is built, it will show Active and have an IP assigned as shown in Figure 8-13 (the IP is partially obscured for security).



☐	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. Now **ssh** into your HMC, run the **vtmenu** command, select the managed system, and find your blank VM name deployed. This will give you the console. Your VM should be retrying to boot from a blank disk. This is shown in Figure 8-14.

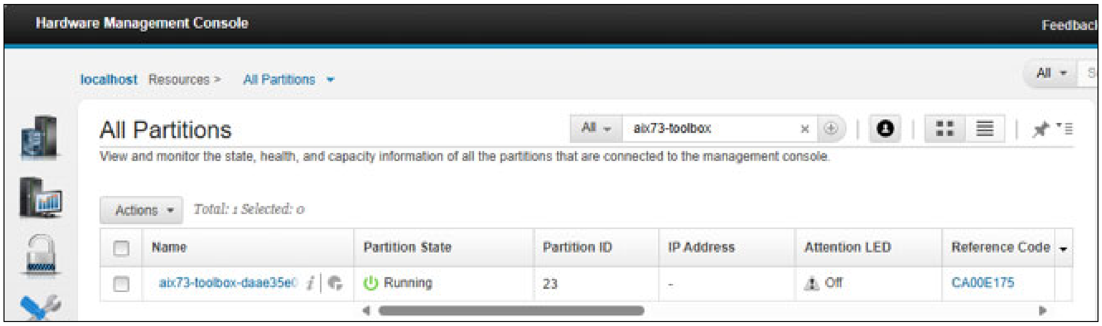


```
 No OS image was detected by firmware.
 At least one disk in the bootlist was not found yet.
 Firmware is now retrying the entries in the bootlist.
 Press ctrl-C to stop retrying.


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/l-lan@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 into each of the VIOS partitions. But first 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.



Hardware Management Console

localhost Resources > All Partitions

All Partitions

View and monitor the state, health, and capacity information of all the partitions that are connected to the management console.

Actions Total: 1 Selected: 0

□	Name	Partition State	Partition ID	IP Address	Attention LED	Reference Code
□	aix73-toolbox-daae35e	Running	23	-	Off	CA00E175

Figure 8-15 Identify the partition ID

10.Convert that to HEX, which in this case is 17. Search for the virtual adapter in the VIOS using the **lsmap** command as shown in Example 8-5.

Example 8-5 VIOS search for the Hex Id of the LPAR

```
$ lsmap -all | grep 00017
vhost17         U9119.MME.xxxxx-V1-C72                     0x00000017
```

11.Create a new file backed optical device, vtopt18, to mount an ISO for the blank VM. Do not use the existing vtopt used by PowerVC to create the blank VM. Create a new one with the VIOS **mkvdev** command as seen in Example 8-6.

Example 8-6 Create new virtual optical device

```
$ 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
$
$ mkvdev -fbo -vadapter vhost17
vtopt18 Available
$  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
Physloc
Mirrored              N/A
```

Note: For more detailed information on creating images for PowerVC please go to the following documentation.
https://www.ibm.com/docs/en/powervc/2.1.1?topic=working-images

12. For mounting the AIX ISO you would need to have an ISO uploaded into the VIOS virtual optical repository. You can do this from the command line or from the HMC.

    a. For the command line use the following document:

       Creating a Virtual Optical Repository on the VIOS using the command line.

    b. To do this from the HMC use the following document:

       Creating the Virtual Optical Repository n the VIOS from the HMC.

    In summary, we copy the ISO to the VIOS and load it into the VIOS virtual optical library using the VIOS **mkvopt** command.

13. We can see the newly available ISO and other existing RHEL8.4 ISO's mounted which are being used by other VMs. This is shown in Example 8-7.

Example 8-7 Optical device created and display of 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. Now that we have a media repository and an AIX ISO available, we can just load it onto the previously created virtual target device (vtopt18) using the **loadopt** command. Then mount the AIX 7.3 ISO to the new VM. This is shown in Example 8-8.

Example 8-8 Mount 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. Now restart the VM and install the operating system from the SMS menu. Boot into SMS Refer to https://www.ibm.com/support/pages/example-using-sms-choose-boot-device for assistance with using the SMS boot menu.
16. Install AIX using the BOS menu (SMS). For more information refer to https://www.ibm.com/docs/en/aix/7.3?topic=system-using-bos-menus.
17. Configure a network to be able to copy the AIX Toolbox ISO to the VM.

Now we want to include the AIX Toolbox in every AIX VM at deploy time. So we add the toolbox to this base image VM we just created.

Install DNF and the AIX Toolbox repository.

Use this blog for the steps to install DNF and the IBM AIX Toolbox repository.

Note that the definition for the remote dnf repository is in /opt/freeware/etc/dnf/dnf.conf.

This file points to the /mnt location that your ISO was mounted on. You can either leave the ISO in the AIX image to be deployed in a filesystem, so that it is copied within each new VM. That would require mounting on /mnt each time you needed to use it. Or update the /opt/freeware/etc/dnf/dnf.conf file to point to a remote yum repository in your organization.

You can create a new yum repo server, by deploying a new RHEL VM using PowerVC.

1. On the yum-server created, run the commands as shown in Example 8-9.

Example 8-9 Install yum repo 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_LCD4107
740.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-server. Update the contents to point to your yum server. This is shown in Example 8-10

Example 8-10 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-server. This is shown in Example 8-11.

Example 8-11 Start httpd on the yum 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. Ensue the yum-server hostname is resolvable in /etc/hosts or a DNS server. If a DNS server, ensure the DNS server is defined in the base VM.

   Now you have a centralized yum server to serve all of the standard VMs you deploy from PowerVC. You no longer need to mount the toolbox ISO locally.
6. You can then install cloud-init on your base AIX VM using dnf as shown in Example 8-12.

Example 8-12 Install cloud-init

```
# dnf install cloud-init
```

   This prepares your AIX VM with the ability to be captured as an OpenStack image.

   This will also ensure that all future VMs deployed from PowerVC using this image are enabled for capture.

7. 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.

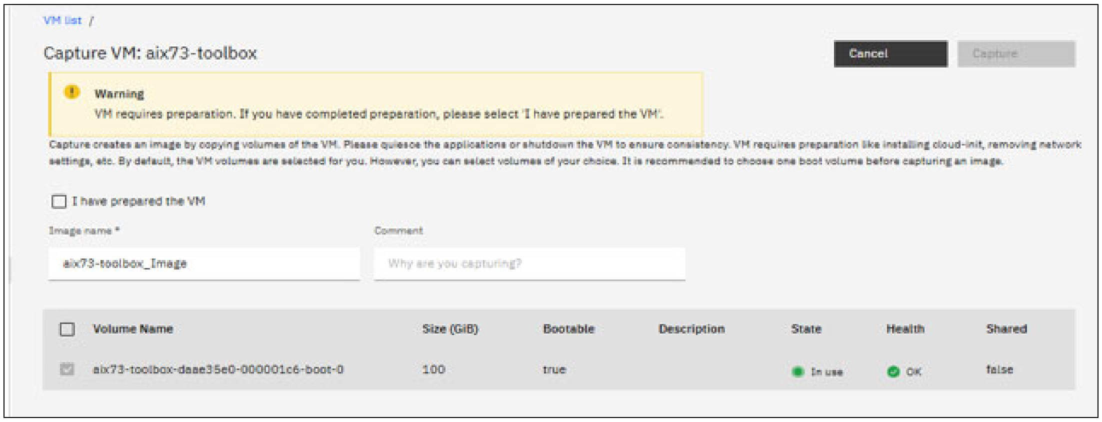
When done, remove the TCP/IP interface from the VM before capture as shown in Example 8-13.

Example 8-13 Remove network interface

```
# ifconfig en0 down detach
```

8. In PowerVC, capture the VM. You will be asked to verify that you have 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.



VM list /

Capture VM: aix73-toolbox

Warning
VM requires preparation. If you have completed preparation, please select 'I have prepared the VM'.

Capture creates an image by copying volumes of the VM. Please quiesce the applications or shutdown the VM to ensure consistency. VM requires preparation like installing cloud-init, removing network settings, etc. By default, the VM volumes are selected for you. However, you can select volumes of your choice. It is recommended to choose one boot volume before capturing an image.

☐ I have prepared the VM

Image name *: aix73-toolbox_Image

Comment: Why are you capturing?

☐	Volume Name	Size (GiB)	Bootable	Description	State	Health	Shared
☑	aix73-toolbox-daae35e0-000001c6-boot-0	100	true		● In use	✓ OK	false

Figure 8-16 Capture the VM in PowerVC

   Now you have an image that can be deployed into a running VM which will have the hostname the same as the VM name specified in PowerVC, and be available to log into with the IP address assigned by PowerVC.

9. Deploy a VM

   Go to images in PowerVC and deploy an instance of the image you just created. Name it mod1. It should deploy a VM with hostname mod1.

Once the image is prepared and captured, we have the ability to deploy a base AIX 7.3 VM in less than 10 minutes from PowerVC, and it includes the IBM AIX ToolBox, ready for modernization tasks on IBM Power Systems.

8.7 Live Update

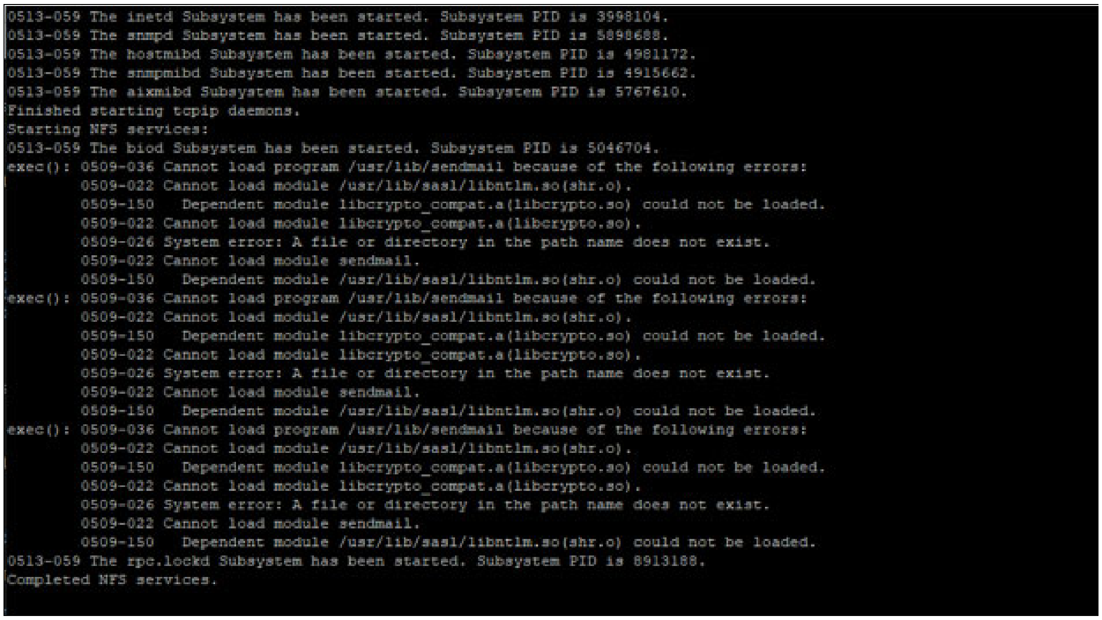
AIX Version 7.2 introduced the AIX Live Update capability, a feature designed to mitigate workload downtime associated with kernel patching. Prior to this, deploying kernel fixes often required a system restart, interrupting running applications. In contrast, Live Update enables the application of interim fixes without halting active workloads, allowing them to immediately utilize the updated code.

IBM delivers kernel patches as interim fixes. In AIX versions prior to 7.2, modifications to the AIX kernel or persistently loaded kernel extensions mandated a host logical partition (LPAR) reboot. While AIX 7.1 and earlier offered concurrent update-enabled interim fixes for a subset of kernel changes, this approach had limitations. Live Update in AIX Version 7.2 extends the principles of concurrent updates, enabling the deployment of a wider range of kernel fixes without the need for a system restart.

8.7.1 Example use of Live Update

For this example we continue with the installed AIX VM mod1 at oslevel 7300-00-00-0000.

We find from starting the VM and displayed on the console, that DNF is installed but some modules are not loading correctly as seen 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 tcpip 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.
```

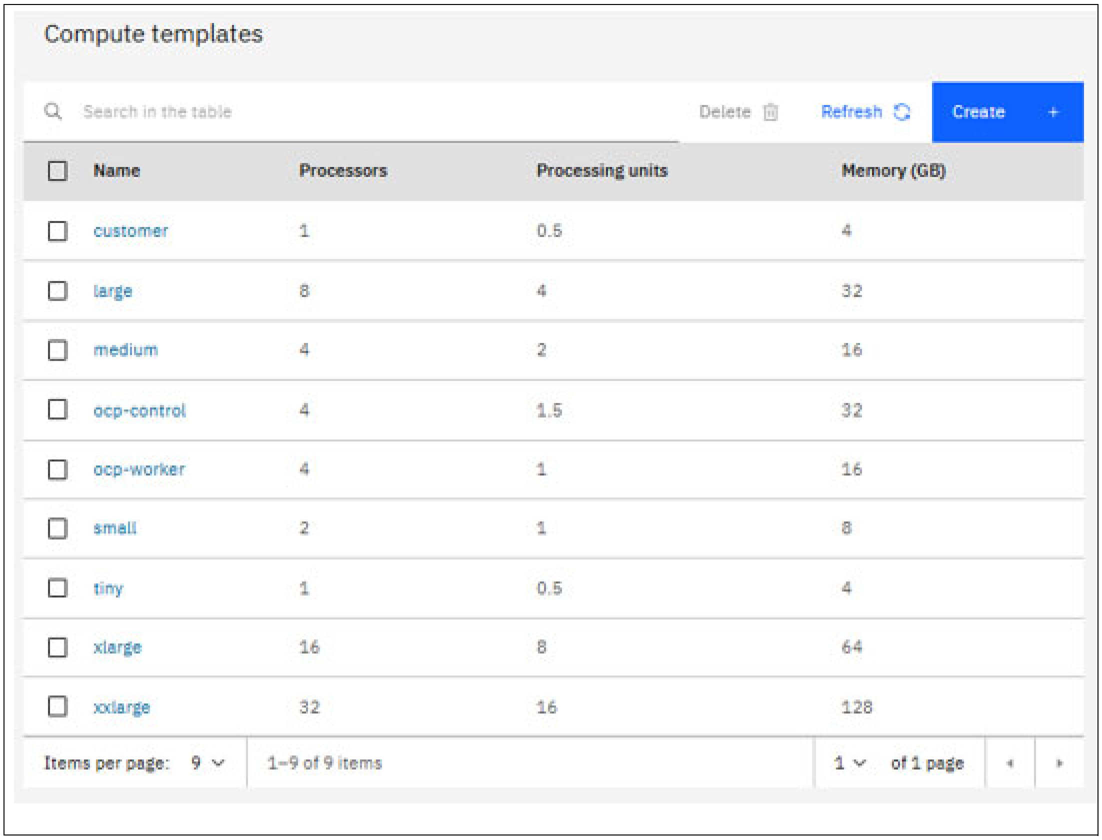
Figure 8-17 New AIX VM installed with DNF at 7300-00 exhibits library load errors

Note: The problem has been identified and described in this technote https://www.ibm.com/support/pages/ibm-aix-sendmail-fails-run-after-upgrading-openssl-30 requires an AIX update.

This issue is fixed with an upgrade to AIX 7.3 TL0 SP3 or higher, or TL1 SP0 or higher.

However, the VM has been handed over to the development teams, who are not aware of the issue as yet. The development teams have waited enough for this VM and do not want any further disruption with reboots.

In addition, when creating our base image in the previous PowerVC deployment tasks, we have used the default compute template, restricting our VM to a maximum of 1 processor. We want to increase this limit without having to shutdown the VM.



Compute templates

□	Name	Processors	Processing units	Memory (GB)
□	customer	1	0.5	4
□	large	8	4	32
□	medium	4	2	16
□	ocp-control	4	1.5	32
□	ocp-worker	4	1	16
□	small	2	1	8
□	tiny	1	0.5	4
□	xlarge	16	8	64
□	xxlarge	32	16	128

Figure 8-18 PowerVC Compute Templates

Live update can help us to correct both these problems.

8.7.2 Live Update Concepts

In the AIX Live Update function, the logical partition (LPAR) where the operation is started is called the original partition. The operation involves another LPAR that is called the surrogate partition. Checkpointing a workload means freezing a running process and saving its current state. Checkpointing processes on an LPAR and restarting them later on another LPAR is called mobility.

The AIX Live Update process is configured by changes in the stanzas of the */var/adm/ras/liveupdate/lvupdate.file* which is derived from the template *lvupdate.template* in the same directory. This update process runs as a singleton, where the **geninstall** command creates a lock file, */usr/lpp/.genlib.lock.check*, to ensure only one instance is running.

The Live Update operation runs in one of the following modes:

- Preview mode

  Preview mode provides an estimation of the total operation time, application blackout time, and an estimation of the resources, for example storage and memory, provided to the user. These estimations are based on the assumption that the surrogate partition has the same resources as the original partition. All the provided inputs are validated and the Live Update limitations are checked.

- Automated mode

  In automated mode, a surrogate partition with the same capacity as the original partition is created, and the original partition is turned off and discarded after the Live Update operation completes.

Further details on Live Update can be found at Live Update Concepts

Live Update Evolution

The Live Update process originally utilized the **emgr** and **installp** command tools to allow for non-disruptive updates of kernel, library and emergency fixes, but now is available for Service pack and technology pack updates. Starting in AIX 7.2, Live Kernel Update started with “fix” updates and has evolved to support for service pack and technology pack updates.
Figure 8-19 shows this evolution.

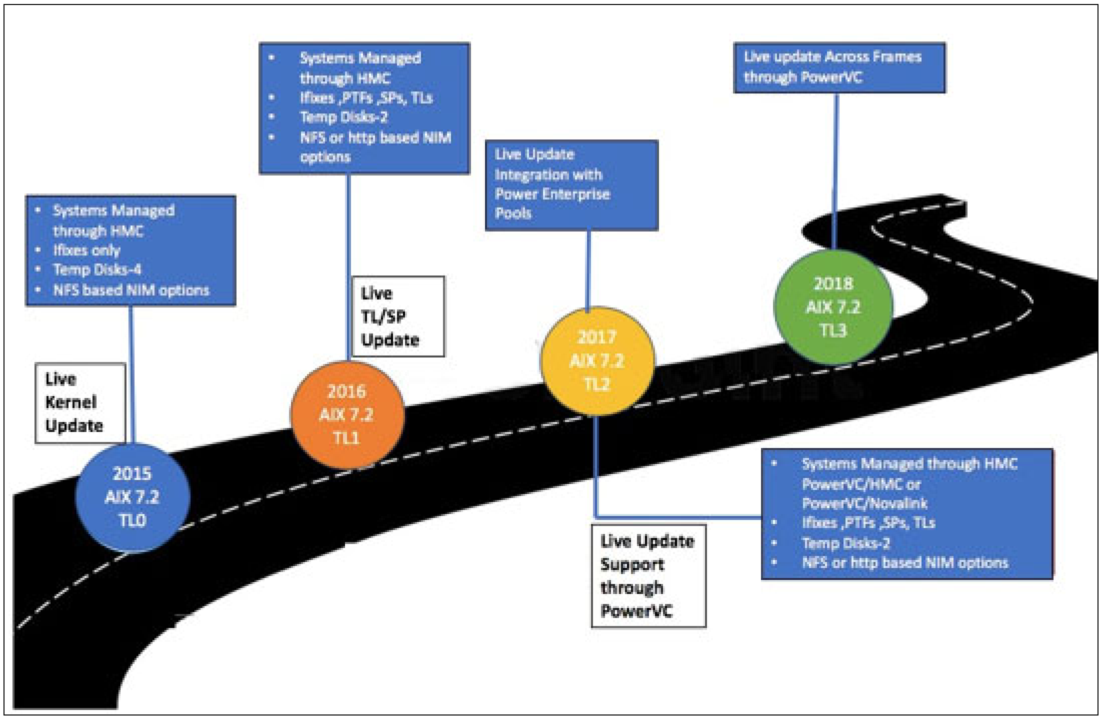

• Systems Managed through HMC
• Ifixes only
• Temp Disks-4
• NFS based NIM options
Live Kernel Update
2015 AIX 7.2 TL0
• Systems Managed through HMC
• Ifixes ,PTFs ,SPs, TLs
• Temp Disks-2
• NFS or http based NIM options
Live TL/SP Update
2016 AIX 7.2 TL1
Live Update Integration with Power Enterprise Pools
2017 AIX 7.2 TL2
Live Update Support through PowerVC
• Systems Managed through HMC PowerVC/HMC or PowerVC/Novalink
• Ifixes ,PTFs ,SPs, TLs
• Temp Disks-2
• NFS or http based NIM options
Live update Across Frames through PowerVC
2018 AIX 7.2 TL3

Figure 8-19 Live update evolution[1]

Live Update

Starting with AIX Version 7.2, the AIX operating system provides the AIX Live Update function. This ability is part of the AIX continuous availability features, that eliminates downtime associated with reboots after applying articular interim fixes and kernel patches.

Concurrent AIX Update uses a method of functional redirection within the in-memory image of the operating system to accomplish patching-in of corrected code. After a fix for a problem has been determined, the corrected code is built, packaged, and tested according to a new process for Concurrent AIX Update. It is then provided to the customer, using the existing interim fix package format.

[1] https://www.ibm.com/support/pages/ibm-aix-72-live-kernel-update-reboot-free-world

Note: The ability to apply or remove a fix without the requirement of a reboot is limited to Concurrent AIX Updates. Technological restrictions prevent some fixes from being made available as a Concurrent AIX Update. In such cases, those fixes may be made available as an interim fix (that is, “traditional” ifix).

Traditional interim fixes for the kernel or kernel extensions still require a reboot of the operating system for both activation and removal.

Live Kernel Update (LKU)

The term Live Update and Live Kernel Update are synonymous, since the process to apply the interim fix is the same. The Kernel wording emphasizes that a kernel interim fix requires a reboot, where a fix applied that does not end with a **bosboot** request s not updating the kernel and does not necessarily require a reboot.

Live Library Update (LLU)

The LLU function shifts the applications from using the old library to the updated new library without any downtime. This is a separate task to the Live Update or Live Kernel Update process.

A library is an entity that provides a set of variables and functions to be used by a program. A library can be an archive or a shared object file. On the AIX operating system, archives can contain both static object files and shared object files as members. In the LLU context, a library denotes a shared object file that is contained in an archive.

The LLU function requires the library to be built as a split library. A library is called split (or LLU-capable) when the shared object file of the library is divided into two separate entities.

You can run the LLU operation by using the **llvupdate** command. An LLU-capable process means that at least one library that is used by the process can be replaced dynamically by using the **llvupdate** command.

- You cannot run the **llvupdate** command separately when a Live Update operation is in progress.
- You cannot run the Live Library Update operation when the **llvupdate** command is running.

For more information on the Live Library Update Process refer to this IBM document.

Live Update Restrictions & Limitations

The following restrictions apply to Live Update:

- Do not change the rootvg, attached devices or filesystems. Do not run any volume group commands or reboot the VIOS, or reboot the PowerVC or HMC during the Live Update Operation.
- After the Live Update operation is complete, if only interim fixes were applied, the mhdisk disk that is specified for the rootvg mirror volume group is labeled as old_rootvg. The old_rootvg volume group can be used for a reboot to 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 filesystems.
- Only supported storage providers in PowerVC are supported with a PowerVC Live Update procedure. Pluggable cinder drivers are not supported.

- IBM Power systems managed by PowerVM Novalink cannot be used along 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.
- You must not start an HMC-based Live Update operation on a partition that is managed by PowerVC because an HMC-based Live Update operation causes issues when PowerVC manages partitions. Unmanage and manage the VM in PowerVC to use an HMC based Live Update.

For a full list of restrictions see Planning Restrictions.

8.7.3 Performing a Live Update using NIM

To perform a Live Update using NIM follow these steps:

1. Create the NIM server

   We start by creating a NIM server at the latest AIX TL 7300-03-00-2446. Mount the ISO for the same oslevel and install the bos.sysmgt.nim.master fileset as shown in Example 8-14.

Example 8-14 Mount ISO for install

```
#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 fileset. as shown in Example 8-15.

Example 8-15 Install the NIM master fileset

```
# installp -gaXcd /mnt/installp/ppc/bos.sysmgt bos.sysmgt.nim.master
```

   For an ethernet network you can use the command shown in Example 8-16 to set up the NIM server. This will create a NIM server that uses the primary interface en0.

Example 8-16 Create the NIM server

```
# nimconfig -a netname=master_net -a pif_name=en0 -a cable_type=tp -v -a
platform=chrp -a  netboot_kernel=mp
```

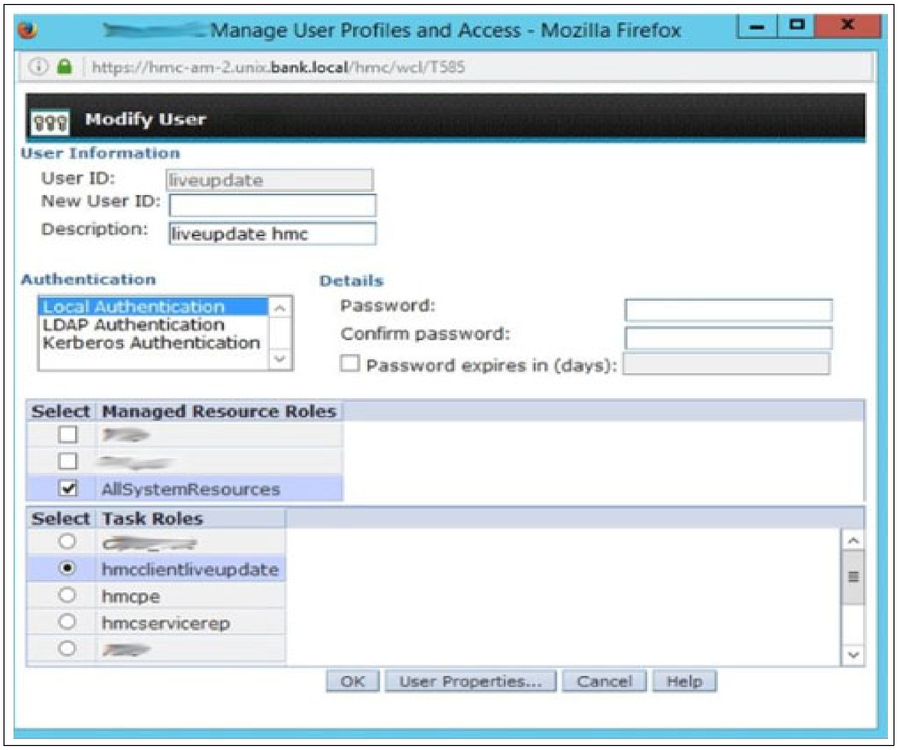
3. Then create the other resources for lpp and spot as given in the procedure documented at this IBM document article.

4. Next, on the NIM server, install dsm.core from the mounted ISO using the command shown in Example 8-17.

Example 8-17 Install dsm.core

```
# installp -gaXcd /mnt/installp/ppc/dsm dsm.core
```

5. Then create the NIM CEC resources for connection to the HMC.
6. Create a user in the HMC named liveupdate. This is shown in Figure 8-20



[redacted] Manage User Profiles and Access - Mozilla Firefox

https://hmc-am-2.unix.bank.local/hmc/wcl/T585

Modify User

User Information

User ID: liveupdate

New User ID:

Description: liveupdate hmc

Authentication

Local Authentication
LDAP Authentication
Kerberos Authentication

Details

Password:

Confirm password:

□ Password expires in (days):

Select	Managed Resource Roles
□	[redacted]
□	[redacted]
☑	AllSystemResources

Select	Task Roles
○	[redacted]
●	hmcclientliveupdate
○	hmcpe
○	hmcservicerep
○	[redacted]

Figure 8-20 Create a user named liveupdate

7. Create an encrypted password file on the NIM server for that user as shown in Example 8-18.

Example 8-18 Create encrypted password file

```
# /usr/bin/dpasswd -f /export/nim/hmc_liveupdate_passwd -U liveupdate
Password:
Re-enter password:
Password file created.
```

8. Define that password file as a NIM object. Replace *$hmc-name* with your DNS or hosts file resolved hmc *hostname*. This is shown in Example 8-19.

Example 8-19 Define password file as an object

```
# nim -o define -t hmc -a if1="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 get the *hscroot* and *liveupdate* users into your known_hosts file of the NIM server. otherwise you will get the error message shown in Example 8-20 when we do the next step shown in Example 8-21 to validates keys. Yes, *hscroot,* not the *liveupdate* user.

Example 8-20 Error message from validating key without correct entry in 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 'xxxxxxxxxxxxxxx' can't be established.
ED25519 key fingerprint is SHA256:P7xlz1Q+K5xPiuWs+XGcDQE++6cLMgFGROkP8liKj+I.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

10. Next, do the key exchange between the NIM server and the HMC as shown in Example 8-22.

Example 8-22 Exchange keys

```
# dkeyexch -f /export/nim/hmc_liveupdate_passwd -I hmc -H $hmc-name
```

You will also need to authorize ssh between the NIM server and the HMC with hmcauth as shown in Example 8-23.

Example 8-23 Authorize ssh

```
# hmcauth -u <liveupdate-user> -p <password> -a <hostname-hmc>
# hmcauth -l
```

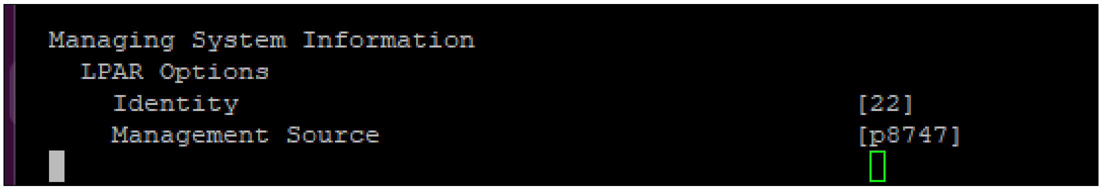
Define the IBM Power Managed systems managed by the HMC as CEC objects in NIM as shown in Example 8-24. Do this for each managed system.

Example 8-24 Define 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
```

11. Now we create NIM definitions for the clients to be subject to a live update.

Use the smit menu, so that you can easily add the lpar id in the information as the Identity, and the management source as the managed system you previously defined. Also define the communication method as nimsh. This is shown in Figure 8-21.



```
Managing System Information
  LPAR Options
    Identity                                              [22]
    Management Source                                     [p8747]
```

Figure 8-21 SMIT menu for adding LPAR definitions

12. Setup **nimsh** on the NIM server, and client using the commands listed in Example 8-25 (the output from the commands is not captured).

Example 8-25 Setup 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
```

13. Test the connection from the NIM server with **`nim -o lslpp mod1.`** This is shown in Example 8-26.

Example 8-26 Test the NIM server connectivity by running a display command

```
root@rbknim:/>nim -o lslpp mod1
  Fileset                      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
```

14. Create a liveupdate resource.The *liveupdate.rte* should exist, which should always be the case for AIX systems running AIX 7.2 or later. This is shown in Example 8-27.

Example 8-27 Create liveupdate resource

```
root@mod1:/> lslpp -L bos.liveupdate.rte
  Fileset                      Level  State  Type  Description (Uninstaller)
  ----------------------------------------------------------------------------
  bos.liveupdate.rte         7.3.0.0    C     F    Live Update Runtime
```

15. Our mod1 VM already has 2 free 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 Copy liveupdate template to liveupdate.data

```
# cp /var/adm/ras/liveupdate/lvupdate.template
/var/adm/ras/liveupdate/lvupdate.data
```

16. Update the contents with the required LPAR information. This is shown in Example 8-29.

Example 8-29 Fill in 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 we were to use the **geninstall -k** command for the liveupdate, the **geninstall** command would refence this *lvupdate.data* file and the location of the emergency fix as shown in Example 8-30.

Example 8-30 Geninstall command

```
# geninstall -k -p -d /tmp/cg/dummy dummy.150813.epkg.Z
```

However, with NIM, we can externalize the *lvupdate.data* file from the VM. Then ensuring there are no details available for a local **geninstall** command on the VM.

17. Define the lvupdate.data as a NIM resources as follows. Copy the original */var/adm/ras/liveupdate/lvupdate.template* which would exist on the NIM server to */export/fixes/lvupdate_data_<LPARNAME>* on the NIM server. Make the same changes for the client as in the previous example except you do not need to add a mode, when using the NIM approach. Instead, the preview is managed by the **installp** preview flag when running the NIM operation.As you will see later.
18. Ensure the value for the management_console in the live_update_data file matches the value returned from the **hmcauth -l** command.

    During the NIM live update task, the NIM defined live update file will copied to the client */var/adm/ras/liveupdate/lvupdate.data* location. The lpar_id specified in the *lvupdate.data* file is a spare lpar_id that will be used to create the surrogate LPAR. This is shown in Example 8-31.

Example 8-31 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 defined for the client on NIM, and can easier manage which clients are targeted for a live update process.

19. Now define the NIM resource type live_update_data as shown in Example 8-32.

Example 8-32 Define 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
```

20. On the client VM mod1 we have three spare disks. We shall use hdisk1 for a traditional alt_disk_backup in case the update process completely fails. The hdisk2 is defined as the ndisk, so will be used to create the surrogate VM. Then hdisk3 is used to mirror the rootvg on the surrogate during the live update process. This is shown in Example 8-33 on page 260.

Example 8-33 Showing available hdisks

```
root@mod1:/tmp>lspv
hdisk0          00cbe757c4603ee5                    rootvg          active
hdisk1          none                                None
hdisk2          none                                None
hdisk3          none                                None
root@mod1:/tmp>
```

21. Let’s take a recent efix for an AIX Security Bulletin:[2]

    `AIX is vulnerable to denial of service (CVE-2024-47102, CVE-2024-52906) (2024.12.24)`

    Download and unpack this into a directory in the */export/efix location* of the NIM server. **`Untar`** the fix file into a directory named *kernext_fix*, and check that it is concurrent enabled with the command shown in Example 8-34.

Example 8-34 Download efix

```
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
LU CAPABLE:       yes
LU CAPABLE:       yes
LU CAPABLE:       yes
LU CAPABLE:       yes
LU CAPABLE:       yes
LU CAPABLE:       yes
LU CAPABLE:       yes
LU CAPABLE:       yes
LU CAPABLE:       yes
LU CAPABLE:       yes
LU CAPABLE:       yes
LU CAPABLE:       yes
LU CAPABLE:       yes
LU CAPABLE:       yes
LU CAPABLE:       yes
LU CAPABLE:       yes
```

22. We define the efix on the NIM server as an lpp_source as shown in Example 8-35.

Example 8-35 Define the efix 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
```

[2] https://www.ibm.com/support/pages/node/7179826?myns=swgother&mynp=OCSWG10&mynp=OCSSPHKW&mync=E&cm_sp=swgother-_-OCSWG10-OCSSPHKW-_-E

```
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:

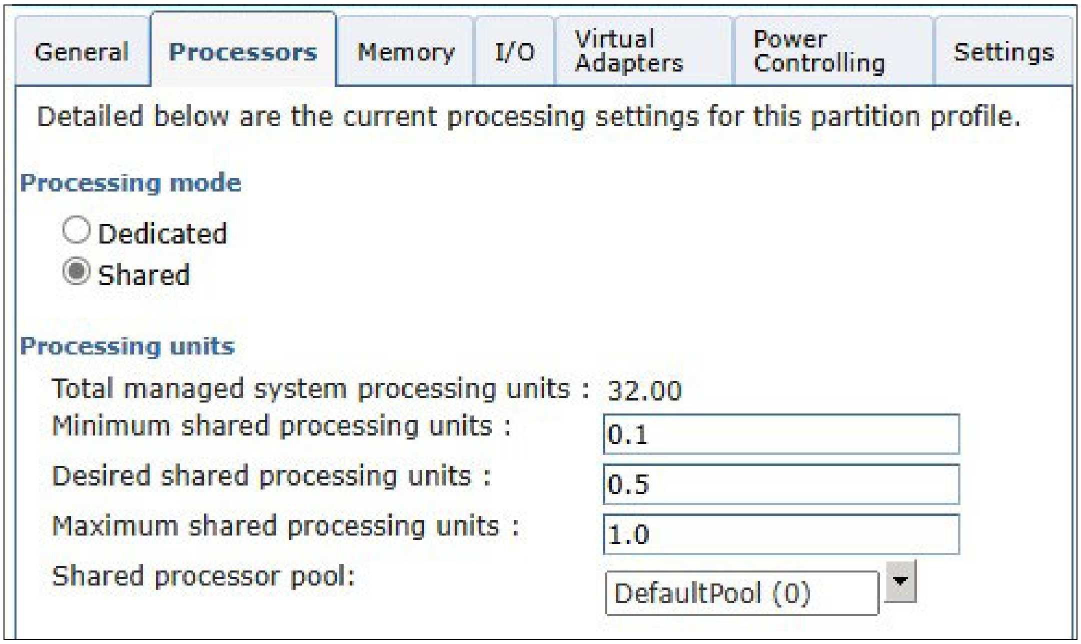
```
warning: 0042-267 c_mk_lpp_source: The defined lpp_source does not have the
"simages" attribute because one or more of the following packages are missing:
```

This warning can be safely ignored. It means that the lpp_source is not bootable because it was created from TL/SP upgrade media rather than AIX base media ISO. SPOT resource can't be created from such lpp_source.

For more details see: https://www.ibm.com/support/pages/how-run-aix-update-nim-clients

23. For this live update, we will also use one of the lpp_source created on the NIM, for 7.3.1.

    To change the max_cpu attribute of the profile we just update the profile before the live update process. The original profile is shown in Figure 8-22.



Detailed below are the current processing settings for this partition profile.

Processing mode

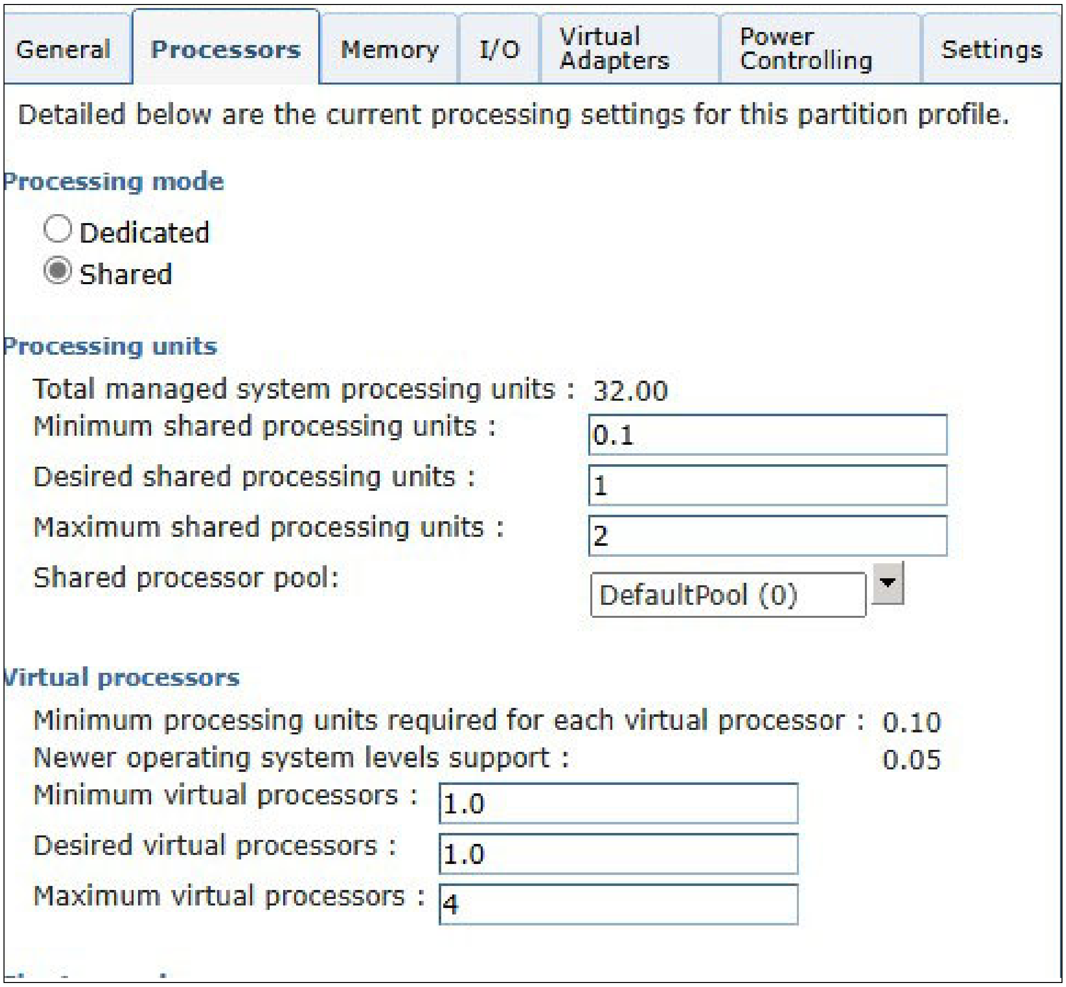
○ Dedicated
● Shared

Processing units

Total managed system processing units : 32.00
Minimum shared processing units : 0.1
Desired shared processing units : 0.5
Maximum shared processing units : 1.0
Shared processor pool: DefaultPool (0)

Figure 8-22 Changing the max_cpu initial view

To the new value we expect to see after the live update process is completed s shown in Figure 8-23



Detailed below are the current processing settings for this partition profile.

Processing mode

○ Dedicated
● Shared

Processing units

Total managed system processing units : 32.00
Minimum shared processing units : 0.1
Desired shared processing units : 1
Maximum shared processing units : 2
Shared processor pool: DefaultPool (0)

Virtual processors

Minimum processing units required for each virtual processor : 0.10
Newer operating system levels support : 0.05
Minimum virtual processors : 1.0
Desired virtual processors : 1.0
Maximum virtual processors : 4

Figure 8-23 Max_cpu value expected after change

Performing the LiveUpdate

Now that we are setup, lets run the LiveUpdate. Perform the following steps:

1. When we run the liveupdate process we get a preview for the TL Update. This is shown in Example 8-36.

Example 8-36 liveupdate preview

```
Results...

SUCCESSES
---------
  Filesets listed in this section passed pre-installation verification
  and will be installed.
Mandatory Fileset Updates
  -------------------------
  (being installed automatically due to their importance)
  bos.rte.install 7.3.1.5                     # LPP Install Commands
Requisites
  ----------
  (being installed automatically;  required by filesets listed above)
  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 filesets mentioned above will be processed after the
       installp update and its requisites are successfully installed.
```

Following this screen the live update preview starts and will give you an estimate of the time for the update. No update will actually occur. at this time.

2. If you get a live update error. Remove the live_update_<node name> nim resource with the command **#nim -Fo remove liveupdate_data_<lpar-name>**.

   Make changes to the live_update_<node name> file on the NIM server, and redefine the file to NIM using **nim -o define -t live_update_data -a server=master -a location=/export/fixes/liveupdate_data_<lpar-name> liveupdate_data_<lpar-name>**

   Then run the preview again. The output should look like Example 8-37.

Example 8-37 LiveUpdate preview successful

```
Verifying 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 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 actually occurred.
********************************************************************************
root@rbknim:/export/fixes>
```

3. When working as expected, run the update without the preview as shown in Example 8-38.

Example 8-38 Run without 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 impending live update.

Creating rootvg for boot of surrogate.
............................................................
Starting the surrogate LPAR.
....................................................................
Creating mirror of original LPAR's rootvg.
................................
Moving workload to surrogate LPAR.
............
        Blackout Time started.

        Blackout Time end.

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 listed in /etc/check_config.files have changed.
        See /var/adm/ras/config.diff for details.
```

The liveupdate process uses dynamic lpar to create a new lpar, so when logging in again you will find that the ssh key has changed.

For further information on the Live Update process refer to the following:

https://www.linkedin.com/pulse/aix-live-update-cookbook-christian-sonnemans/
http://gibsonnet.net/blog/cgaix/html/html/AIX%20Live%20Update%20using%20NIM.html
http://gibsonnet.net/blog/cgaix/html/resource/AIXLiveUpdateblog.pdf

9

IBM i

Modernizing the IBM i environment is a comprehensive process that goes far beyond simply updating user interfaces. It requires evolving existing applications and infrastructure to better align with current business requirements and technological innovations. This includes enhancing application architecture, optimizing data management, and integrating with cutting-edge technologies like web services and mobile apps. A key motivator behind IBM i modernization is the desire to retain the platform’s strong, reliable core while expanding its capabilities to meet the needs of the digital era.

A significant part of IBM i modernization is transforming legacy code, such as RPG, into more contemporary and maintainable formats. This often involves migrating to RPG Free Form, which enhances readability and compatibility with other programming languages. Database modernization is also vital, focusing on standardizing data access through SQL and boosting database performance. Additionally, modernization efforts often involve revamping user interfaces, creating web-based or mobile-friendly front ends for a more intuitive and accessible user experience. Ultimately, the aim is to preserve the critical business logic and data within IBM i systems while ensuring their seamless integration into today’s modern IT ecosystems.

The following topics are covered in this chapter.

© Copyright IBM Corp. 2025.

9.1 IBM i Modernization

The IBM i platform has long served as a reliable and robust foundation for critical business applications across numerous organizations. Renowned for its integrated architecture and strong performance, it continues to underpin essential operations. However, the contemporary business and technological landscape is in constant flux, necessitating that these established systems evolve to meet new demands.

The sphere of IBM i modernization encompasses several key dimensions.

- Application Modernization focuses on updating the user interface, often transitioning from the traditional “green screen” to more intuitive graphical user interfaces (GUIs). It also involves refactoring or re-architecting existing code, such as the conversion of RPG to the more flexible free-form RPG. A crucial aspect is the integration of IBM i applications with newer systems through the use of Application Programming Interfaces (APIs), and the adoption of modern software development practices, most notably DevOps.
- Infrastructure Modernization entails upgrading the physical hardware, for example, to the latest IBM Power10 systems, virtualizing the operating environment, embracing cloud or hybrid cloud deployment models, and implementing automation to enhance operational efficiency.
- Data Modernization centers on improving data management practices, bolstering security measures, and enhancing data accessibility. This includes integrating data across disparate systems to create a unified information landscape, and leveraging data for advanced analytics and business intelligence purposes.

It is important to recognize that IBM i modernization can be approached incrementally, allowing organizations to make progress in manageable stages rather than undertaking a disruptive, all-encompassing “big bang” overhaul. The understanding that modernization is a flexible and adaptable journey, rather than a fixed destination, is paramount for successful implementation.

IBM i modernization is a comprehensive process of evolving existing IBM i applications to integrate with modern technologies and meet current business demands. This includes leveraging current investments while adopting new tools and architectural approaches. Modernization aims to enhance several critical areas, such as improving how customers interact with applications, boosting the efficiency of development teams, strengthening system security, and increasing overall performance. It also involves adapting applications for cloud environments, offering more flexibility and scalability.

A primary driver for IBM i modernization is the desire to enhance user experiences and improve developer productivity. Organizations are also looking to bolster security, achieve better performance, and integrate their IBM i applications with newer technologies and cloud infrastructures. Modernization efforts are focused on key areas such as enhancing the user interface from traditional green screens to more modern graphical interfaces. Organizations also aim to improve data accessibility, gain better insights through integration with analytics, and ensure their systems can scale to support future business growth. Addressing the skills gap by making systems more accessible to developers with contemporary skills is another significant driver.

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 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 non-negotiable driver. Legacy systems can harbor security vulnerabilities, and modernization allows organizations to implement up-to-date security measures and ensure adherence to evolving regulatory requirements.

Modernizing IBM i can lead to several key benefits. These include the ability to leverage existing IT investments, enhanced security measures, improved customer experiences through modern interfaces, and increased productivity for development teams by providing them with better tools and environments. Modernization also brings increased agility, allowing for faster development and deployment of new features. Furthermore, updating user interfaces and improving performance leads 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 variety of strategies and approaches tailored to their specific needs and objectives. The journey typically begins with a thorough assessment and planning phase, involving a detailed analysis of the existing application portfolio and infrastructure to pinpoint modernization requirements and develop a strategic roadmap.

One common approach is UI Modernization, also known as screen refacing, which focuses on transforming the traditional character-based “green screen” interfaces into more modern and intuitive web-based or graphical user interfaces (GUIs). For developers working directly on the platform, IBM Rational Developer for i offers an integrated development environment (IDE) built on the widely used Eclipse framework, specifically designed for creating and maintaining applications on IBM i systems.

Another key area is Code Modernization, which can involve several techniques. IBM i Modernization Engine for Lifecycle Integration (Merlin) stands out as a dedicated application development and modernization environment for IBM i users. Running within Red Hat OpenShift containers, Merlin provides a set of tools designed to guide software developers in modernizing IBM i applications, offering features for RPG conversion, streamlining the DevOps pipeline, and enabling cloud integration.

RPG Conversion automates the process of converting legacy fixed-format RPG code to the more contemporary and flexible free-form RPG. Enhancing traditional programming languages is discussed in section 9.5, “Traditional Programming languages” on page 306.

In some cases, organizations may opt for Language Migration, which involves rewriting applications entirely in modern programming languages such as Java, PHP, Python, or .NET. We discuss the use of modern open-source languages in section 9.6, “Open-Source Programming on IBM i” on page 308. Code Refactoring is another strategy that focuses on restructuring and optimizing existing code to enhance performance and maintainability without altering its external functionality.

Database Modernization is crucial for leveraging the full potential of the IBM i system. This often involves converting older data constructs to create a fully relational database, which improves performance and makes the database more understandable for developers familiar with modern database concepts. DB2 for IBM i continues to evolve to support new data requirements within IBM i. We discuss these new capabilities in section 9.3, “DB2 for i” on page 285.

9.2 Modern development environments on IBM i

The modernization of application development on IBM i, while promising, faces a significant hurdle: a growing skills shortage. As traditional RPG and COBOL developers approach retirement, the pool of professionals proficient in these legacy languages is shrinking. Simultaneously, while modern languages like Python and Java are gaining traction on the platform, there's a need to bridge the gap between these new technologies and the unique IBM i environment. This scarcity of developers with the right blend of legacy and modern skills poses a challenge for businesses seeking to modernize their IBM i applications, potentially slowing down digital transformation efforts and impacting their ability to innovate.

IBM has introduced new development environments for IBM i programmers to help clients maintain and modernize their legacy applications, minimize the training required for IBM i developers, and bridge the skills gap.

This section discusses two of those options, the use of Visual Studio Code and the use of Merlin.

9.2.1 Visual Studio Code as a development tool for IBM i

In this section we will be detailing the option of using Visual Studio Code as a development tool, also known as an IDE (Integrated Development Environment).

Over time, IBM has provided several development tools for IBM i. However many of these are outdated and are either withdrawn or no longer supported. These include, but are not limited to:

- Report Layout Utility (RLU)
- Screen Design Aid (SDA)
- File Compare and Merge Utility (FCMU)
- Advanced Printer Function (AFP)
- Character Generator Utility (CGU)
- Data File Utility (DFU)

For more detail on the status of these products see `https://www.ibm.com/docs/en/announcements/end-marketing-end-support-i-merlin-10-rdi-96-selected-i-functions?region=US`

IBM i does have a modern and supported IDE, Visual Studio Code.

Introduction to Visual Studio Code?

VS Code was developed by Microsoft for Windows in 2015, as a light version of its commercial Visual Studio product and released as open source. Visual Studio Code, also known as VSCode, is a cross-platform, multi-language code editor which can be used for modern day development projects on IBM i.

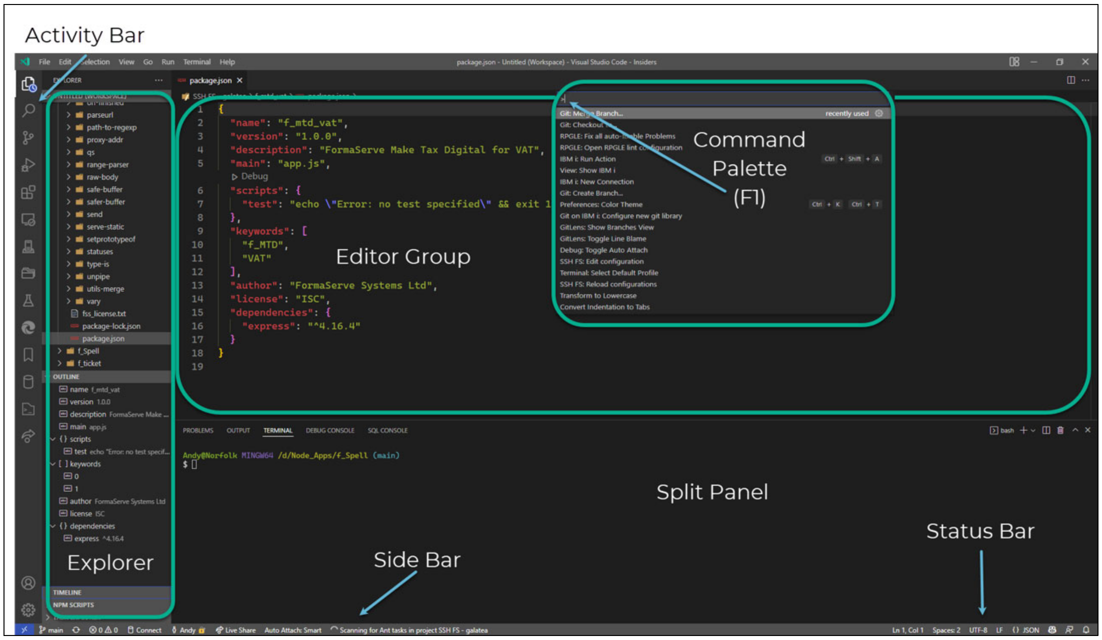
You can find information about Visual Studio Code at this link. The Visual Studio Code GitHub repository is located at `https://github.com/microsoft/vscode.` All examples, and screenshots, shown in this topic, are produced using Microsoft Windows 11 running Windows Sub-system for Linux (WSL) with Ubuntu 24.04.5 LTS.

Installing Visual Studio Code

Visual Studio Code can be installed on various operating systems. These are listed on the Microsoft installation page of Visual Studio Code which can be found on the install page for Visual Studio Code. There is also an insider's version of Visual Studio Code, where the latest features are released on a daily cycle. If you are interested in this version of Visual Studio Code, details can be found at the Visual Studio Code insider's page.

Note: Visual Studio Code is installed locally on your PC, or Mac, not on your IBM Power server.

The layout of Visual Studio Code is like any modern-day IDE, whether that be Eclipse, NetBeans or other similar products. Figure 9-1 shows the layout of Visual Studio Code. If you are coming from an IBM Rational Developer for i Eclipse (RDi) based background, this should be familiar to you.



```
{
  "name": "f_mtd_vat",
  "version": "1.0.0",
  "description": "FormaServe Make Tax Digital for VAT",
  "main": "app.js",
  ▷ Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1
  },
  "keywords": [
    "f_MTD",
    "VAT"
  ],
  "author": "FormaServe Systems Ltd",
  "license": "ISC",
  "dependencies": {
    "express": "^4.16.4"
  }
}
```

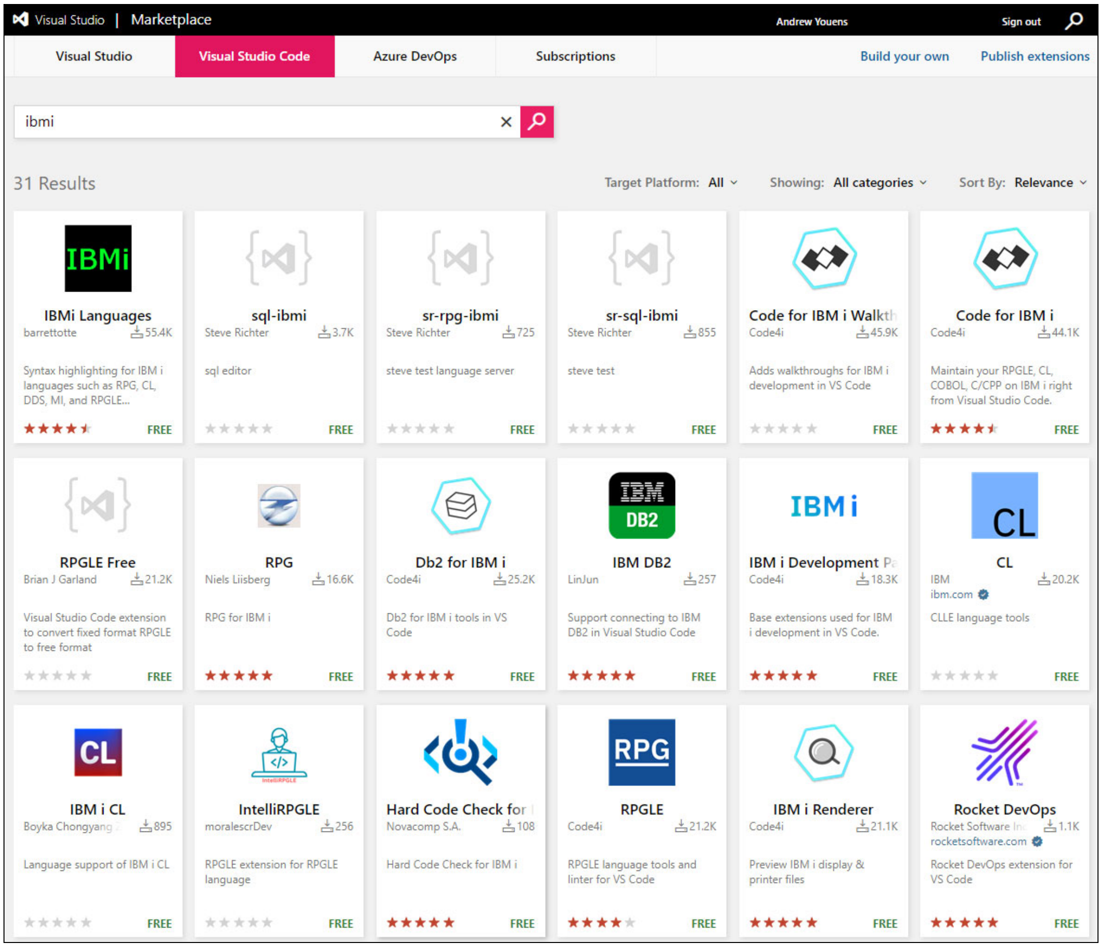
```
Andy@Norfolk MINGW64 /d/Node_Apps/f_Spell (main)
$ 
```

Figure 9-1 VS Code layout

Visual Studio Code's initial launch presents a deliberately minimalist interface. This "lean" architecture signifies that the base installation provides a focused suite of tools optimized for core source code editing and management within prevalent open-source programming languages. Functionality beyond this core set is implemented via an extensive extension ecosystem.

For instance, developers requiring customized themes or specific keybindings (e.g., replicating a Notepad++ configuration) can utilize the rich extension marketplace. This repository offers a diverse selection of extensions, ranging from UI customizations and keymap configurations, to language-specific support, debugging capabilities, and integrated development tools. With a current catalog exceeding 69,000 extensions, developers possess granular control over their VS Code environment, enabling a highly tailored and performant workflow.

Figure 9-2 on page 270 shows a subset of the catalog, filtered by entering "IBM" in the search bar.



Visual Studio | Marketplace — Andrew Youens — Sign out

Visual Studio | Visual Studio Code | Azure DevOps | Subscriptions — Build your own — Publish extensions

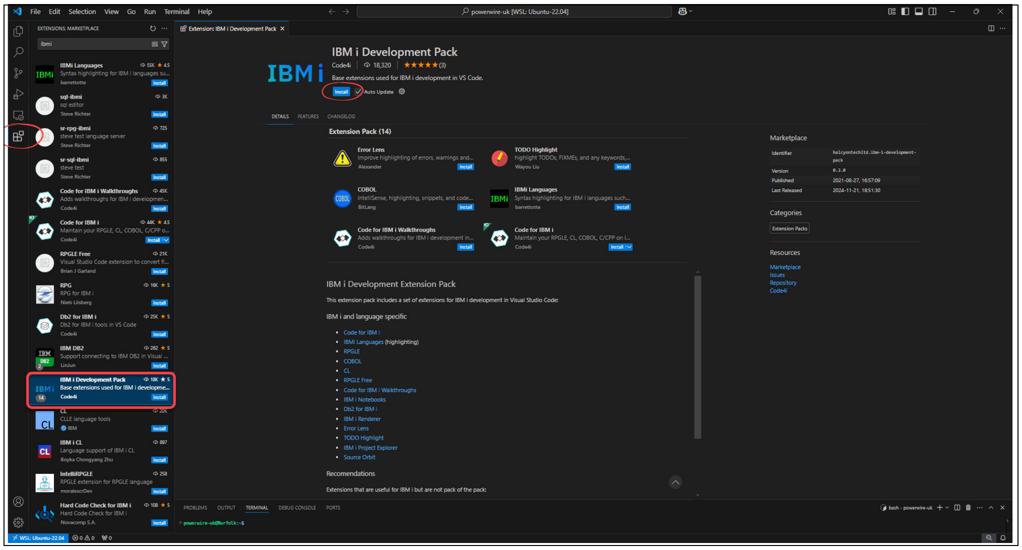
ibmi

31 Results — Target Platform: All — Showing: All categories — Sort By: Relevance

Extension	Publisher	Installs	Description	Price
IBMi Languages	barrettotte	55.4K	Syntax highlighting for IBM i languages such as RPG, CL, DDS, MI, and RPGLE...	FREE
sql-ibmi	Steve Richter	3.7K	sql editor	FREE
sr-rpg-ibmi	Steve Richter	725	steve test language server	FREE
sr-sql-ibmi	Steve Richter	855	steve test	FREE
Code for IBM i Walkth	Code4i	45.9K	Adds walkthroughs for IBM i development in VS Code	FREE
Code for IBM i	Code4i	44.1K	Maintain your RPGLE, CL, COBOL, C/CPP on IBM i right from Visual Studio Code.	FREE
RPGLE Free	Brian J Garland	21.2K	Visual Studio Code extension to convert fixed format RPGLE to free format	FREE
RPG	Niels Liisberg	16.6K	RPG for IBM i	FREE
Db2 for IBM i	Code4i	25.2K	Db2 for IBM i tools in VS Code	FREE
IBM DB2	LinJun	257	Support connecting to IBM DB2 in Visual Studio Code	FREE
IBM i Development P	Code4i	18.3K	Base extensions used for IBM i development in VS Code.	FREE
CL	IBM ibm.com	20.2K	CLLE language tools	FREE
IBM i CL	Boyka Chongyang	895	Language support of IBM i CL	FREE
IntelliRPGLE	moralescrDev	256	RPGLE extension for RPGLE language	FREE
Hard Code Check for	Novacomp S.A.	108	Hard Code Check for IBM i	FREE
RPGLE	Code4i	21.2K	RPGLE language tools and linter for VS Code	FREE
IBM i Renderer	Code4i	21.1K	Preview IBM i display & printer files	FREE
Rocket DevOps	Rocket Software In rocketsoftware.com	1.1K	Rocket DevOps extension for VS Code	FREE

Figure 9-2 Visual studio extensions

To get started with IBM i development, it is recommended to install the IBM i development pack extension. This extension includes the starting blocks necessary for IBM i development with Visual Studio Code. This is shown in Figure 9-3.



powerwire-uk [WSL: Ubuntu-22.04]

Extension: IBM i Development Pack

IBM i Development Pack

Code4i | 18,320 | ★★★★★ (3)

Base extensions used for IBM i development in VS Code.

Install ☑ Auto Update

DETAILS FEATURES CHANGELOG

Extension Pack (14)

Extension	Description	Publisher
Error Lens	Improve highlighting of errors, warnings and...	Alexander
TODO Highlight	highlight TODOs, FIXMEs, and any keywords,...	Wayou Liu
COBOL	IntelliSense, highlighting, snippets, and code...	BitLang
IBMi Languages	Syntax highlighting for IBM i languages such...	barrettotte
Code for IBM i Walkthroughs	Adds walkthroughs for IBM i development in...	Code4i
Code for IBM i	Maintain your RPGLE, CL, COBOL, C/CPP on I...	Code4i

IBM i Development Extension Pack

This extension pack includes a set of extensions for IBM i development in Visual Studio Code:

IBM i and language specific

- Code for IBM i
- IBMi Languages (highlighting)
- RPGLE
- COBOL
- CL
- RPGLE Free
- Code for IBM i Walkthroughs
- IBM i Notebooks
- Db2 for IBM i
- IBM i Renderer
- Error Lens
- TODO Highlight
- IBM i Project Explorer
- Source Orbit

Recomendations

Extensions that are useful for IBM i but are not pack of the pack:

Marketplace

Identifier	halcyontechltd.ibm-i-development-pack
Version	0.2.0
Published	2021-08-27, 16:57:09
Last Released	2024-11-21, 18:51:30

Categories

Extension Packs

Resources

Marketplace
Issues
Repository
Code4i

File Edit Selection View Go Run Terminal Help
EXTENSIONS: MARKETPLACE
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS

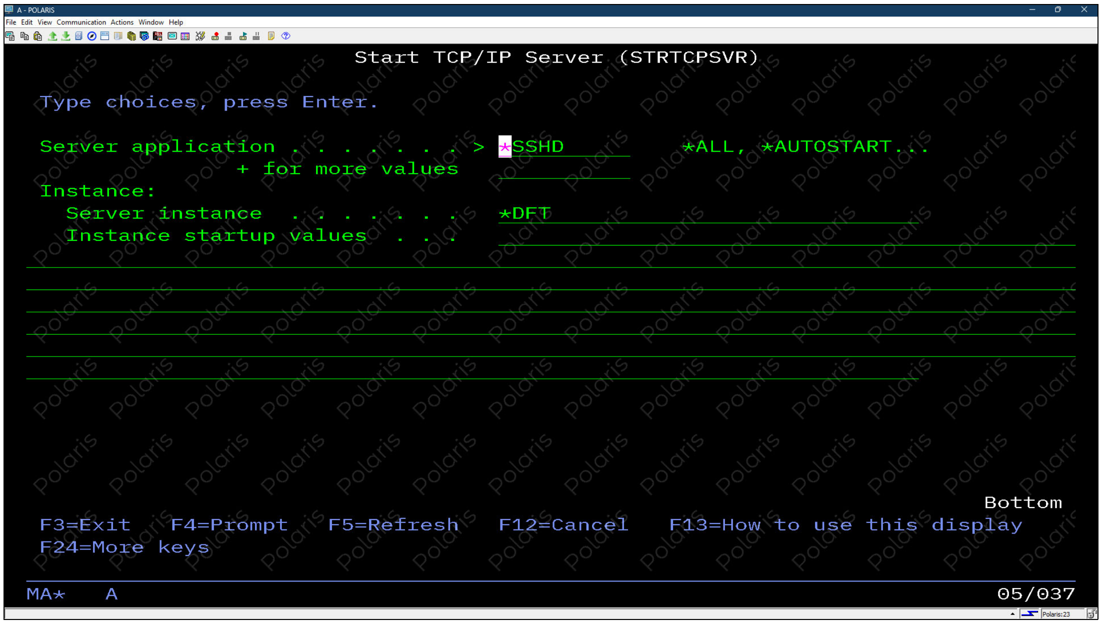
Figure 9-3 IBM i development pack

IBM i Configuration for VS Code

Once you have installed VS Code on your desktop, the next step is to configure a connection to the IBM i system. This process creates a SSH secure connection between your desktop to the 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 command STRTCPSVR (Start TCP/IP Server). See Figure 9-4. It is also possible to start this server using IBM i Navigator (iNav).



```
                        Start TCP/IP Server (STRTCPSVR)

  Type choices, press Enter.

  Server application . . . . . . . > *SSHD          *ALL, *AUTOSTART...
                 + for more values
  Instance:
    Server instance  . . . . . . .   *DFT
    Instance startup values  . . .

                                                                        Bottom
  F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
  F24=More keys
 MA*   A                                                                05/037
```

Figure 9-4 Starting the SSH service

Once the extension has been installed, we notice a new icon in the activity bar of your VS Code instance as shown in Figure 9-5.

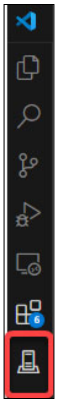
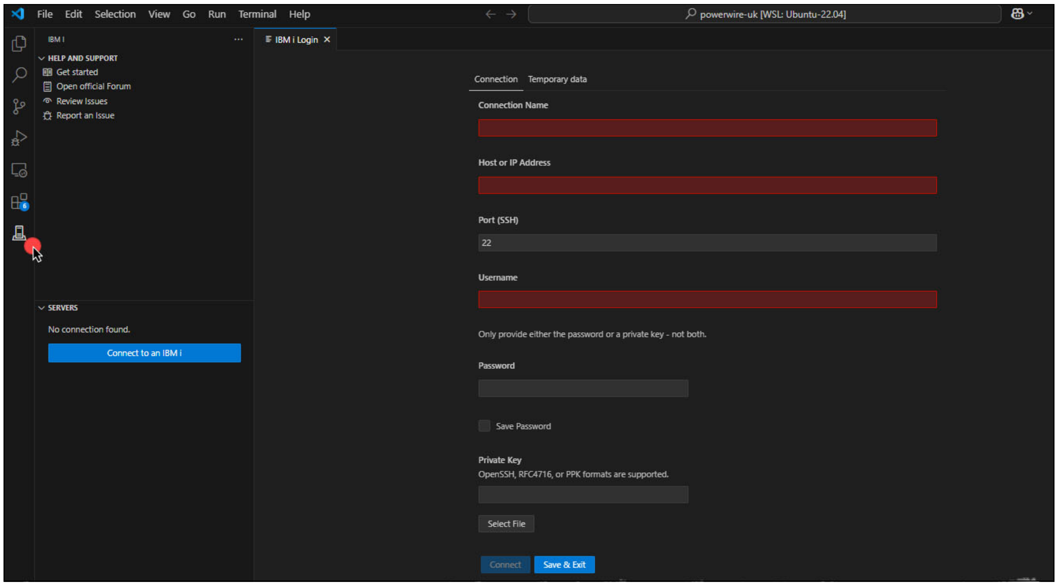


Figure 9-5 Connection configuration tool

Selecting this icon will open the settings to configure our connection to the IBM i LPAR. Enter the specific connection details in the configuration panel as shown in Figure 9-6.



Connection Name

Host or IP Address

Port (SSH)
22

Username

Only provide either the password or a private key - not both.

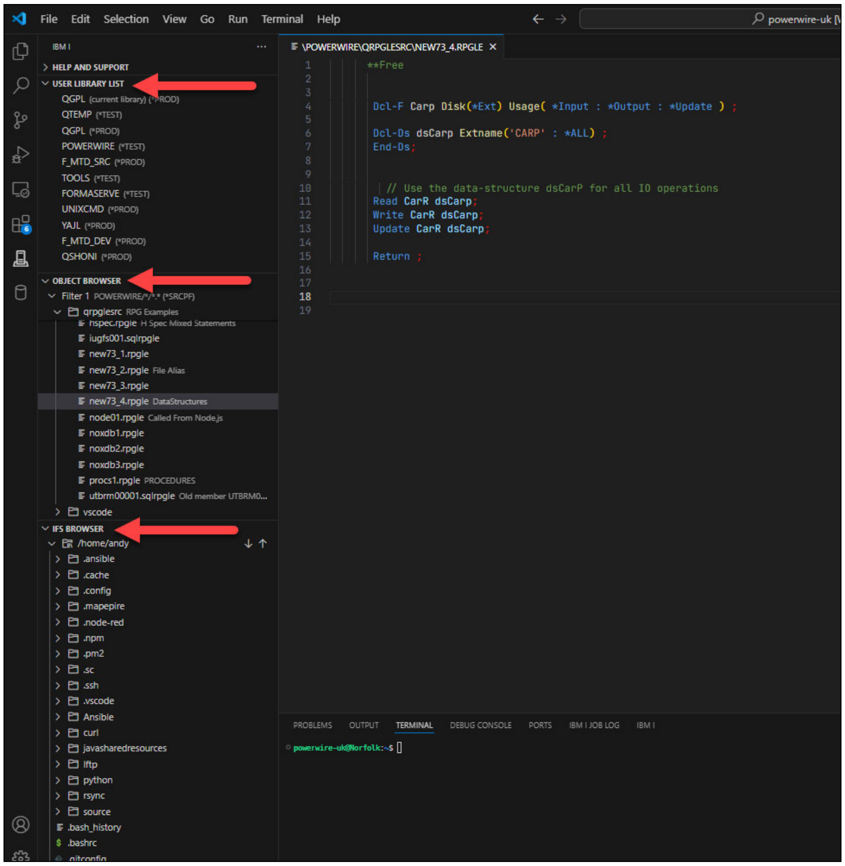
Password

□ Save Password

Private Key
OpenSSH, RFC4716, or PPK formats are supported.

Figure 9-6 Entering details for connection

Once a connection has been established, we can see all our library list, objects and IFS directories as seen in Figure 9-7



```
**Free


 Dcl-F Carp Disk(*Ext) Usage( *Input : *Output : *Update ) ;

 Dcl-Ds dsCarp Extname('CARP' : *ALL) ;
 End-Ds;


   // Use the data-structure dsCarP for all IO operations
 Read CarR dsCarp;
 Write CarR dsCarp;
 Update CarR dsCarp;

 Return ;
```

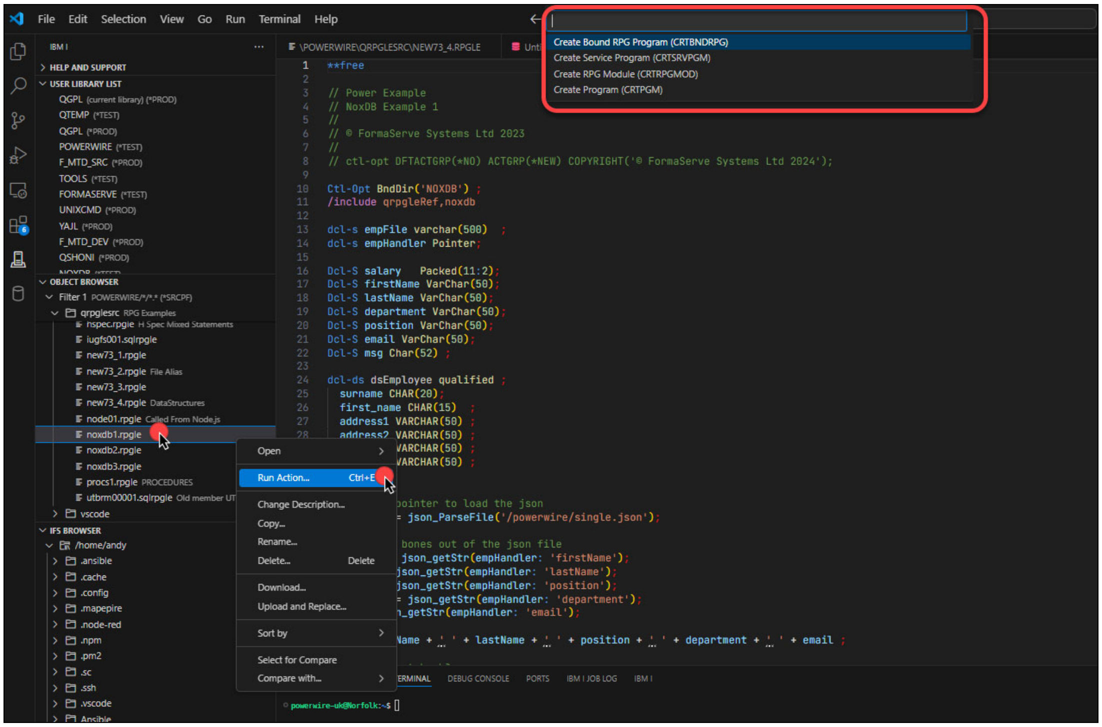
```
powerwire-uk@Norfolk:~$
```

Figure 9-7 Visual Studio after connecting to IBM i LPAR

Opening source code and compiling

Opening source code, either to edit or browse, is as simple as finding your source in either the object browser, or IFS browser and then double clicking on it to open it in the editor window.

Once you have found your source code and you are ready to compile it, right click on the source in the explorer. Then take the option to run Action (Ctrl+E), which will bring up the valid options for the source in the command window box, at the top of the screen. This is shown in Figure 9-8.

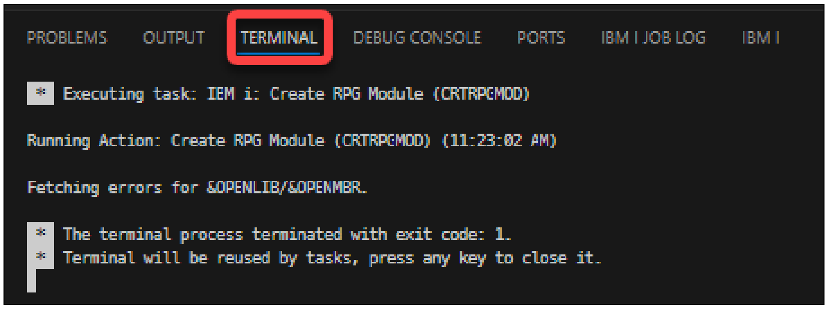


```
**free

// Power Example
// NoxDB Example 1
//
// © FormaServe Systems Ltd 2023
//
// ctl-opt DFTACTGRP(*NO) ACTGRP(*NEW) COPYRIGHT('© FormaServe Systems Ltd 2024');

Ctl-Opt BndDir('NOXDB') ;
/include qrpgleRef,noxdb

dcl-s empFile varchar(500)  ;
dcl-s empHandler Pointer;

Dcl-S salary   Packed(11:2);
Dcl-S firstName VarChar(50);
Dcl-S lastName VarChar(50);
Dcl-S department VarChar(50);
Dcl-S position VarChar(50);
Dcl-S email VarChar(50);
Dcl-S msg Char(52) ;

dcl-ds dsEmployee qualified ;
  surname CHAR(20);
  first_name CHAR(15)  ;
  address1 VARCHAR(50) ;
  address2 VARCHAR(50) ;
          VARCHAR(50) ;
          VARCHAR(50) ;


          pointer to load the json
          = json_ParseFile('/powerwire/single.json');

           bones out of the json file
           json_getStr(empHandler: 'firstName');
          json_getStr(empHandler: 'lastName');
          json_getStr(empHandler: 'position');
          = json_getStr(empHandler: 'department');
          n_getStr(empHandler: 'email');

          Name + ' ' + lastName + ' ' + position + ' ' + department + ' ' + email ;
```

```
powerwire-uk@Norfolk:~$ 
```

Figure 9-8 Select action

Compile messages, including compile failure, will be shown in the tabbed terminal window as shown in Figure 9-9.



```
 *  Executing task: IBM 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 Compiler messages

Clicking on any error will show the line the compile failed on.

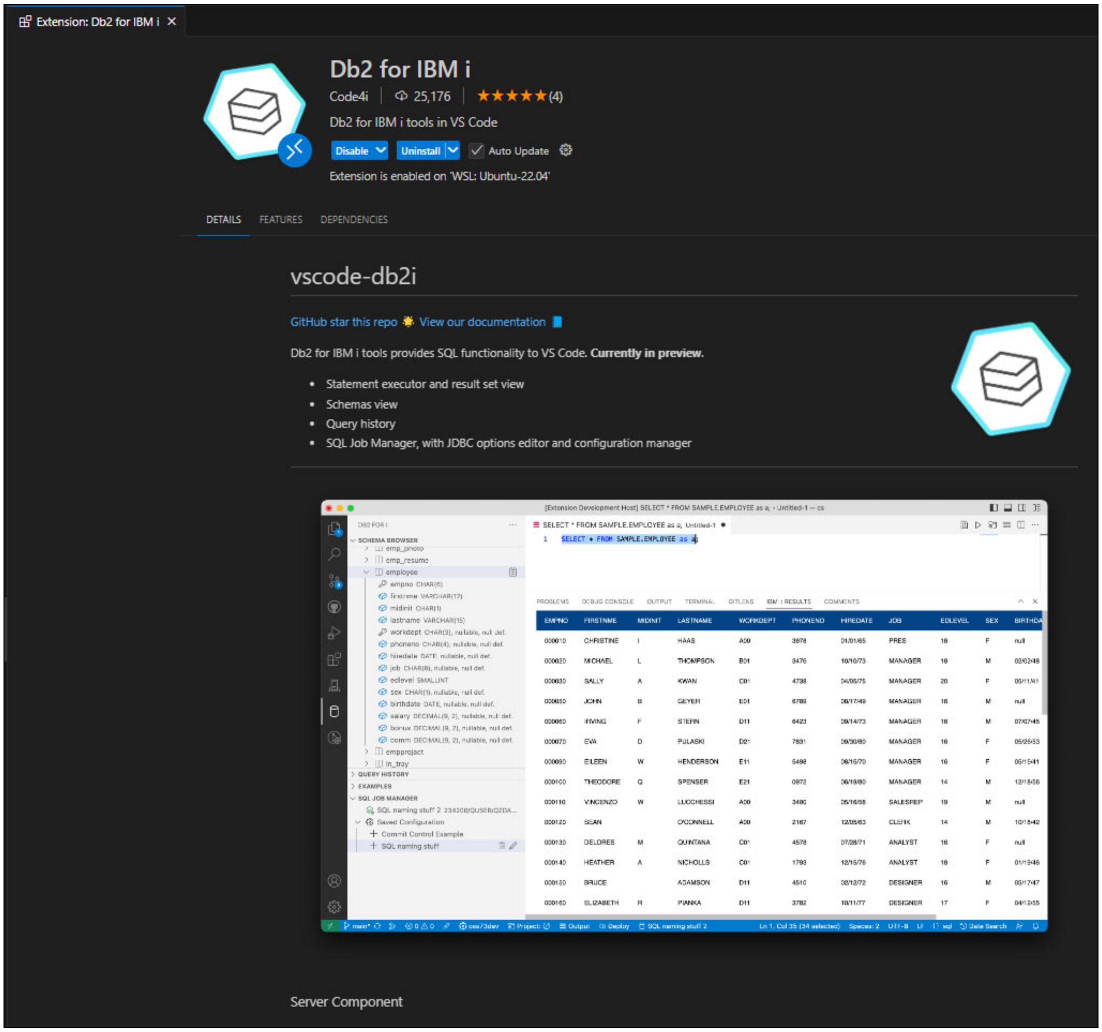
DB2 for i Extension

The DB2 for i extension allows the running of SQL statements, which gives the option of performing any SQL statement.

The extension is installed, as with the Code for i extension, through the Visual Studio Marketplace from the following link:

https://marketplace.visualstudio.com/items?itemName=HalcyonTechLtd.vscode-db2i

It can also be installed using the extension button in the Visual Studio Code activity bar. as shown in Figure 9-10. This extension is also part of the IBM i development pack and will be automatically installed with that option.



Db2 for IBM i

Code4i | 25,176 | ★★★★★ (4)

Db2 for IBM i tools in VS Code

Extension is enabled on 'WSL: Ubuntu-22.04'

vscode-db2i

GitHub star this repo 🌟 View our documentation 📘

Db2 for IBM i tools provides SQL functionality to VS Code. **Currently in preview.**

- Statement executor and result set view
- Schemas view
- Query history
- SQL Job Manager, with JDBC options editor and configuration manager

[Extension Development Host] SELECT * FROM SAMPLE.EMPLOYEE as a; - Untitled-1 — cs

DB2 FOR I

SCHEMA BROWSER
- emp_photo
- emp_resume
- employee
  - empno CHAR(6)
  - firstnme VARCHAR(12)
  - midinit CHAR(1)
  - lastname VARCHAR(15)
  - workdept CHAR(3), nullable, null def.
  - phoneno CHAR(4), nullable, null def.
  - hiredate DATE, nullable, null def.
  - job CHAR(8), nullable, null def.
  - edlevel SMALLINT
  - sex CHAR(1), nullable, null def.
  - birthdate DATE, nullable, null def.
  - salary DECIMAL(9, 2), nullable, null def.
  - bonus DECIMAL(9, 2), nullable, null def.
  - comm DECIMAL(9, 2), nullable, null def.
- empproject
- in_tray

QUERY HISTORY

EXAMPLES

SQL JOB MANAGER
- SQL naming stuff 2 234208/QUSER/QZDA...
- Saved Configuration
  - Commit Control Example
  - SQL naming stuff

SELECT * FROM SAMPLE.EMPLOYEE as a; Untitled-1

```sql
SELECT * FROM SAMPLE.EMPLOYEE as a;
```

EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDA
000010	CHRISTINE	I	HAAS	A00	3978	01/01/65	PRES	18	F	null
000020	MICHAEL	L	THOMPSON	B01	3476	10/10/73	MANAGER	18	M	02/02/48
000030	SALLY	A	KWAN	C01	4738	04/05/75	MANAGER	20	F	05/11/41
000050	JOHN	B	GEYER	E01	6789	08/17/49	MANAGER	16	M	null
000060	IRVING	F	STERN	D11	6423	09/14/73	MANAGER	16	M	07/07/45
000070	EVA	D	PULASKI	D21	7831	09/30/80	MANAGER	16	F	05/26/53
000090	EILEEN	W	HENDERSON	E11	5498	08/15/70	MANAGER	16	F	05/15/41
000100	THEODORE	Q	SPENSER	E21	0972	06/19/80	MANAGER	14	M	12/18/56
000110	VINCENZO	W	LUCCHESSI	A00	3490	05/16/58	SALESREP	19	M	null
000120	SEAN		O'CONNELL	A00	2167	12/05/63	CLERK	14	M	10/18/42
000130	DELORES	M	QUINTANA	C01	4578	07/28/71	ANALYST	16	F	null
000140	HEATHER	A	NICHOLLS	C01	1793	12/15/76	ANALYST	18	F	01/19/46
000150	BRUCE		ADAMSON	D11	4510	02/12/72	DESIGNER	16	M	05/17/47
000160	ELIZABETH	R	PIANKA	D11	3782	10/11/77	DESIGNER	17	F	04/12/55

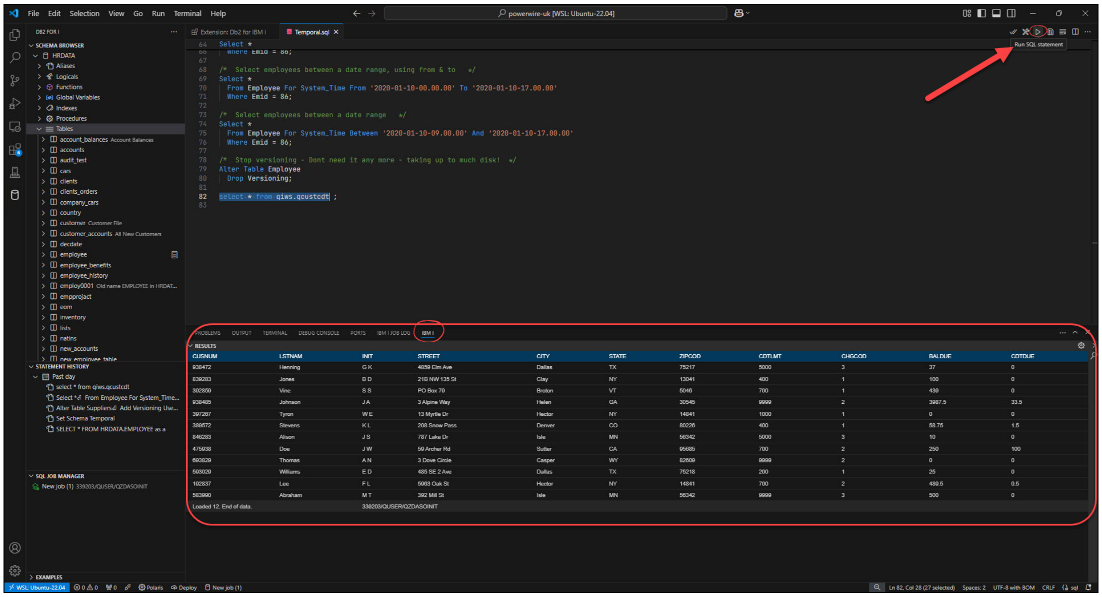
main* 0 0 cos/3dev Project: Output Deploy SQL naming stuff 2 Ln 1, Col 35 (34 selected) Spaces: 2 UTF-8 LF sql Data Search

Server Component

Figure 9-10   Db2 for IBM i extension

To execute a SQL statement, either open an existing .sql file or create a new one. This will open it within the editor window. Once you are ready to execute the SQL code, either select the **run** button in the top right corner, or use the shortcut key of **CTRL-R**. Any resultset or errors, will be shown in the IBM i tab.

Figure 9-11 shows this.



```
Select *
  Where Emid = 86;

/*  Select employees between a date range, using from & to   */
Select *
  From Employee For System_Time From '2020-01-10-00.00.00' To '2020-01-10-17.00.00'
  Where Emid = 86;

/*  Select employees between a date range   */
Select *
  From Employee For System_Time Between '2020-01-10-09.00.00' And '2020-01-10-17.00.00'
  Where Emid = 86;

/*  Stop versioning - Dont need it any more - taking up to much disk!  */
Alter Table Employee
  Drop Versioning;

select * from qiws.qcustcdt ;
```

CUSNUM	LSTNAM	INIT	STREET	CITY	STATE	ZIPCOD	CDTLMT	CHGCOD	BALDUE	CDTDUE
938472	Henning	G K	4859 Elm Ave	Dallas	TX	75217	5000	3	37	0
839283	Jones	B D	21B NW 135 St	Clay	NY	13041	400	1	100	0
392859	Vine	S S	PO Box 79	Broton	VT	5046	700	1	439	0
938485	Johnson	J A	3 Alpine Way	Helen	GA	30545	9999	2	3987.5	33.5
397267	Tyron	W E	13 Myrtle Dr	Hector	NY	14841	1000	1	0	0
389572	Stevens	K L	208 Snow Pass	Denver	CO	80226	400	1	58.75	1.5
846283	Alison	J S	787 Lake Dr	Isle	MN	56342	5000	3	10	0
475938	Doe	J W	59 Archer Rd	Sutter	CA	95685	700	2	250	100
693829	Thomas	A N	3 Dove Circle	Casper	WY	82609	9999	2	0	0
593029	Williams	E D	485 SE 2 Ave	Dallas	TX	75218	200	1	25	0
192837	Lee	F L	5963 Oak St	Hector	NY	14841	700	2	489.5	0.5
583990	Abraham	M T	392 Mill St	Isle	MN	56342	9999	3	500	0

Loaded 12. End of data. 339203/QUSER/QZDASOINIT

Figure 9-11 Screen with resultset displayed

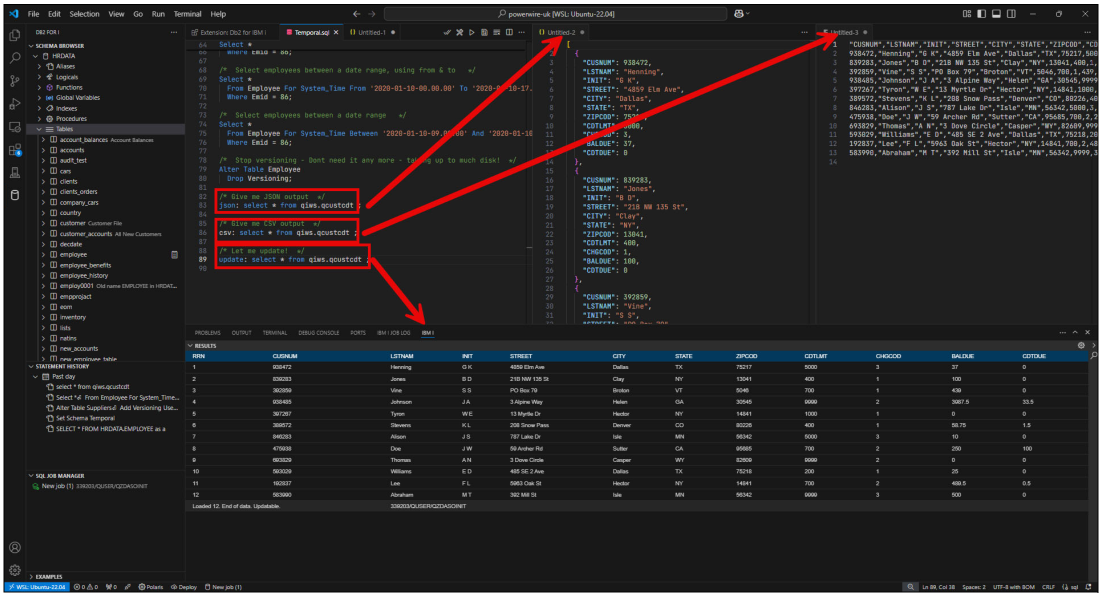
it is also possible to format the output of the SQL statement result set by prefixing the SQL with any of the following options:

- – JSON
- – CSV
- – Update

This is done by specifying the format you want the results in followed by a colon in front of the SQL statement. For example to get a resultset in JSON format, use the following command

```
JSON: select * from hrdata.employee ;
```

These three options are shown in Figure 9-12.



```
Select *
  Where Emid = 86;

/*  Select employees between a date range, using from & to   */
Select *
  From Employee For System_Time From '2020-01-10-00.00.00' To '2020-01-10-17.
  Where Emid = 86;

/*  Select employees between a date range   */
Select *
  From Employee For System_Time Between '2020-01-10-09.00.00' And '2020-01-10
  Where Emid = 86;

/*  Stop versioning - Dont need it any more - taking up to much disk!  */
Alter Table Employee
  Drop Versioning;

/* Give me JSON output  */
json: select * from qiws.qcustcdt ;

/* Give me CSV output  */
csv: select * from qiws.qcustcdt ;

/* Let me update!  */
update: select * from qiws.qcustcdt ;
```

```
[
  {
    "CUSNUM": 938472,
    "LSTNAM": "Henning",
    "INIT": "G K",
    "STREET": "4859 Elm Ave",
    "CITY": "Dallas",
    "STATE": "TX",
    "ZIPCOD": 75217,
    "CDTLMT": 5000,
    "CHGCOD": 3,
    "BALDUE": 37,
    "CDTDUE": 0
  },
  {
    "CUSNUM": 839283,
    "LSTNAM": "Jones",
    "INIT": "B D",
    "STREET": "21B NW 135 St",
    "CITY": "Clay",
    "STATE": "NY",
    "ZIPCOD": 13041,
    "CDTLMT": 400,
    "CHGCOD": 1,
    "BALDUE": 100,
    "CDTDUE": 0
  },
  {
    "CUSNUM": 392859,
    "LSTNAM": "Vine",
    "INIT": "S S",
```

```
"CUSNUM","LSTNAM","INIT","STREET","CITY","STATE","ZIPCOD","CD
938472,"Henning","G K","4859 Elm Ave","Dallas","TX",75217,500
839283,"Jones","B D","21B NW 135 St","Clay","NY",13041,400,1,
392859,"Vine","S S","PO Box 79","Broton","VT",5046,700,1,439,
938485,"Johnson","J A","3 Alpine Way","Helen","GA",30545,9999
397267,"Tyron","W E","13 Myrtle Dr","Hector","NY",14841,1000,
389572,"Stevens","K L","208 Snow Pass","Denver","CO",80226,40
846283,"Alison","J S","787 Lake Dr","Isle","MN",56342,5000,3,
475938,"Doe","J W","59 Archer Rd","Sutter","CA",95685,700,2,2
693829,"Thomas","A N","3 Dove Circle","Casper","WY",82609,999
593029,"Williams","E D","485 SE 2 Ave","Dallas","TX",75218,20
192837,"Lee","F L","5963 Oak St","Hector","NY",14841,700,2,48
583990,"Abraham","M T","392 Mill St","Isle","MN",56342,9999,3
```

RRN	CUSNUM	LSTNAM	INIT	STREET	CITY	STATE	ZIPCOD	CDTLMT	CHGCOD	BALDUE	CDTDUE
1	938472	Henning	G K	4859 Elm Ave	Dallas	TX	75217	5000	3	37	0
2	839283	Jones	B D	21B NW 135 St	Clay	NY	13041	400	1	100	0
3	392859	Vine	S S	PO Box 79	Broton	VT	5046	700	1	439	0
4	938485	Johnson	J A	3 Alpine Way	Helen	GA	30545	9999	2	3987.5	33.5
5	397267	Tyron	W E	13 Myrtle Dr	Hector	NY	14841	1000	1	0	0
6	389572	Stevens	K L	208 Snow Pass	Denver	CO	80226	400	1	58.75	1.5
7	846283	Alison	J S	787 Lake Dr	Isle	MN	56342	5000	3	10	0
8	475938	Doe	J W	59 Archer Rd	Sutter	CA	95685	700	2	250	100
9	693829	Thomas	A N	3 Dove Circle	Casper	WY	82609	9999	2	0	0
10	593029	Williams	E D	485 SE 2 Ave	Dallas	TX	75218	200	1	25	0
11	192837	Lee	F L	5963 Oak St	Hector	NY	14841	700	2	489.5	0.5
12	583990	Abraham	M T	392 Mill St	Isle	MN	56342	9999	3	500	0

Loaded 12. End of data. Updatable. 339203/QUSER/QZDASOINIT

Figure 9-12 JSON formatting of SQL outputs

To execute a CL command from within your SQL script, prefix the CL statement with **`cl:`** as shown in the following line.

```
cl: crtlib mylib
```

Once the SQL is executed, the create library command will be run.

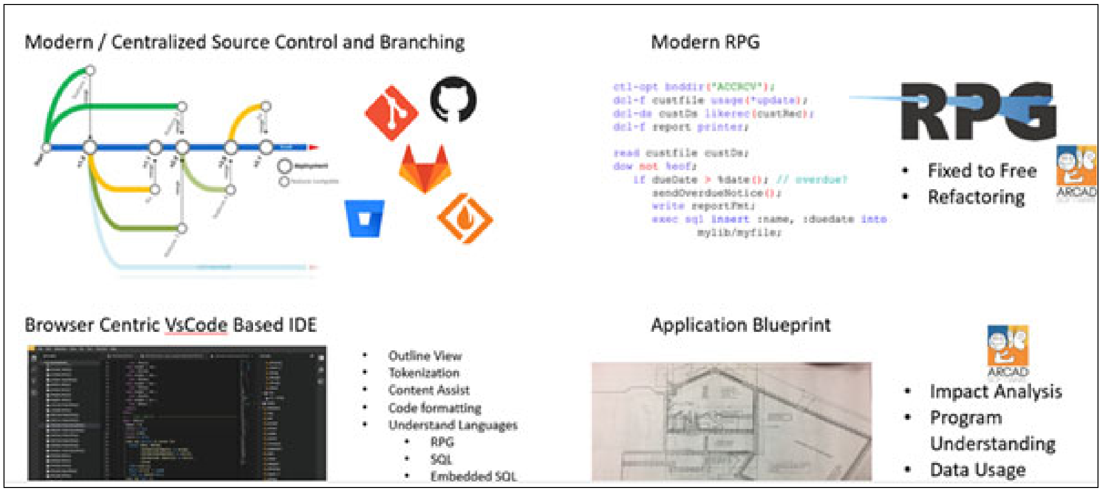
9.2.2 Merlin

The IBM i platform continues to be a cornerstone for businesses worldwide, but the landscape of application development is rapidly evolving. To thrive in this dynamic environment, IBM i development must embrace modern practices and tools. This is where Merlin comes in.

Merlin, a shortcut for IBM i **M**odernization **E**ngine fo**R** **L**ifecycle **I**ntegratio**N**, is a comprehensive solution designed to modernize the IBM i development experience. At its core, Merlin aligns IBM i application development with industry standards, providing support for tools like Jenkins for continuous integration, Git for source control management, and the browser-based VSCode (Visual Studio Code) for a modern development environment. This integration streamlines the development process, fostering collaboration, automation, and efficiency.

Beyond streamlining development, Merlin also provides key modernization capabilities. A critical aspect of this is the automated conversion of fixed-format RPG to free-format RPG. The different version of RPG are discussed in 9.5.1, “RPG” on page 306. This feature helps to bring legacy codebases up to modern standards, improving readability, maintainability, and developer productivity. Merlin also includes in-depth application impact analysis, enabling developers to understand the potential consequences of code changes before they are implemented. This reduces the risk of errors and simplifies the modernization process.

An overview of Merlin’s capabilities is shown in Figure 9-13.



```
ctl-opt bnddir('ACCRCV');
dcl-f custfile usage(*update);
dcl-ds custDs likerec(custRec);
dcl-f report printer;

read custfile custDs;
dow not %eof;
   if dueDate > %date(); // overdue?
      sendOverdueNotice();
      write reportFmt;
      exec sql insert :name, :duedate into
            mylib/myfile;
```

Figure 9-13 Merlin feature overview.

Merlin benefits

For organizations invested in IBM i, Merlin doesn't replace existing tools like Rational Developer for i (RDi) - it complements them. Developers now have choices: continue with workstation-based RDi or embrace the browser-accessible, container-based Merlin experience. Both paths remain fully supported, with Merlin offering additional integrated capabilities through partnerships with solution providers like Arcad.

Merlin distinguishes itself by offering a fully integrated and supported suite of tools, encompassing an IDE, plugins, and utilities that enable modern development practices. This includes seamless integration with CI/CD pipelines, native Git-based source control, and application impact analysis. Notably, Merlin also integrates ARCAD Software's DevOps expertise, incorporating support for standard tools like Git and Jenkins, and includes ARCAD functions, further enhancing its modernization capabilities.

Merlin makes it also easier for teams to provision test environments for modernization by leveraging IBM PowerVS and PowerVC in a simplified and automated way. Instead of requiring deep technical skills or manual setup, Merlin streamlines the process-allowing users to spin up environments quickly and consistently through guided workflows. This is especially helpful for teams looking to experiment or validate changes in a safe, repeatable setup without needing to become infrastructure experts. By reducing the complexity behind the scenes, Merlin helps organizations focus more on innovation and less on configuration. Figure 9-14 provides a general overview of Merlin.

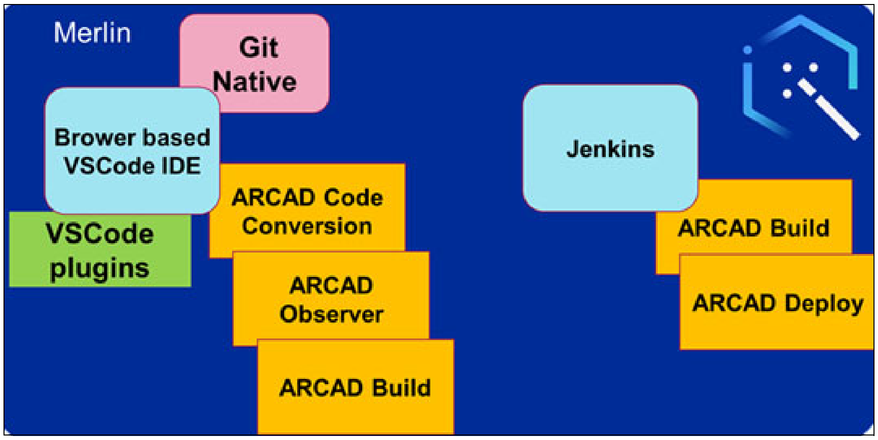


Figure 9-14 General view of Merlin

Issues Merlin Helps Solve in IBM i Development

IBM i Merlin addresses several critical challenges that traditional IBM i development environments face in today's rapidly evolving technology landscape. By bringing modern development practices to this established platform, Merlin bridges the gap between IBM i's robust foundation and contemporary software development approaches.

Modern & Centralized Source Control and Branching

Traditional IBM i development often relied on library-based source management systems that lacked the sophisticated versioning capabilities modern developers expect. Merlin solves this by natively integrating with Git, enabling:

- Complete source code history with detailed tracking of who changed what and when
- Support for concurrent development through branching and merging
- Proper isolation of feature development from production code
- Collaborative workflows where multiple developers can work on the same codebase without conflicts
- Simplified code reviews and approval processes before changes reach production

This integration transforms IBM i development from isolated, sequential work to a collaborative, parallel process that significantly improves productivity and code quality.

Modern RPG Development

The evolution from fixed-format to free-format RPG represents a major modernization challenge for many IBM i shops. Merlin addresses this with:

- Automated conversion tools that transform legacy fixed-format RPG into modern free-format code
- Syntax highlighting and code completion that supports the latest RPG syntax
- Refactoring tools that help developers adopt modern programming practices
- Consistency checks to ensure code follows modern standards
- Support for the latest RPG features that fixed-format cannot leverage

Merlin provides impact analysis that enhances code comprehension and maintainability by visually mapping dependencies and potential change effects across the application. This modernization path preserves the business logic while making code more maintainable and accessible to new developers who expect modern language features.

Browser-Centric VS Code-Based IDE

Traditional development required installing and maintaining workstation software. Merlin's browser-based approach delivers significant advantages:

- Access to development environments from any device with a browser
- Consistent development experience regardless of location or hardware
- Reduced workstation management overhead for IT departments
- Merlin's browser-based environment reduces the need for local installs, minimizing security risks and ensuring centralized control.
- Simplified onboarding for new developers with familiar VS Code interface
- Extensions ecosystem compatibility for customizing the development experience
- Real-time collaboration features not possible with desktop tools

Its modern interface is designed to feel natural for existing IBM i developers, eliminating the need for extensive retraining. This accessibility transforms development from a location-dependent activity to something that can happen anywhere, anytime, with minimal setup.

Application Blueprint and Impact Analysis

Understanding complex IBM i applications has traditionally been challenging. Merlin's application blueprint capabilities provide:

- Visual representation of application components and their relationships
- Impact analysis that shows exactly what will be affected by code changes
- Dependency mapping to understand the full scope of modifications
- Risk assessment before making changes to critical systems
- Knowledge preservation as application understanding moves from tribal knowledge to documented structures

CI/CD Integration

Traditional IBM i development often involved manual processes for moving code through development, testing, and production environments. This approach was time-consuming, error-prone, and difficult to standardize across teams. Merlin solves this fundamental challenge by providing seamless integration with Jenkins and modern DevOps pipelines.

When developers use Merlin, their code changes can automatically trigger build processes, test suites, and deployment procedures. For example, when a developer commits code to a specific branch, Jenkins can automatically compile the RPG code, run unit tests, and prepare the objects for deployment. This automation dramatically reduces human error and ensures consistency in how code moves through environments.

The CI/CD capabilities in Merlin also enable practices like continuous testing, where automated test suites verify that new changes don't break existing functionality. This creates a safety net that allows developers to make changes with greater confidence, knowing that problems will be caught early in the development cycle rather than in production. This approach is shown in Figure 9-15.

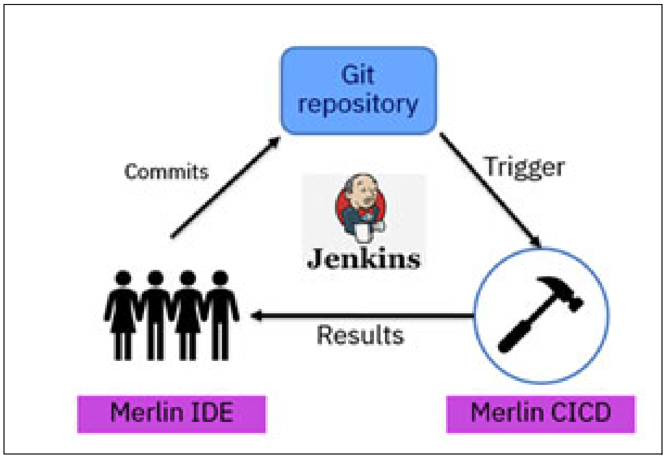


Figure 9-15 CI/CD for IBM

Environment Provisioning

IBM i developers have traditionally faced significant waiting periods when they needed new development or test environments. Setting up these environments often required manual intervention from system administrators, leading to delays and reduced productivity.

Merlin's container-based architecture transforms this process by enabling on-demand creation of development and test environments. When a developer needs a fresh environment to test a new feature or reproduce a bug, they can provision one in minutes rather than days or weeks. These environments are consistent and reproducible, eliminating the “it works on my machine” problem that often plagues development teams.

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 have sometimes been 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 OpenShift.

This flexibility allows organizations to align their IBM i development practices with broader cloud strategies. For example, a company might maintain their production IBM i workloads on-premises while moving development and testing to the cloud, taking advantage of elastic resources and reduced capital expenditure.

The cloud compatibility also enables global development teams to collaborate more effectively, accessing development environments from anywhere without requiring complex VPN setups or remote desktop solutions. This capability is particularly valuable for organizations with distributed teams or those embracing remote work policies.

Skills Gap

The IBM i platform has traditionally required specialized knowledge that's becoming increasingly scarce as experienced developers retire. This creates a significant skills gap that threatens the sustainability of IBM i applications.

Merlin addresses this challenge by providing familiar, modern tools that appeal to developers with contemporary skills. The VS Code-compatible IDE means that developers already familiar with web and open-source development can quickly become productive with IBM i without extensive retraining.

The modernization tools in Merlin also help transform legacy code into more maintainable, understandable formats that align with current programming practices. This transformation makes it easier for new developers to understand and modify existing applications, reducing the dependence on specialized knowledge held by a small group of experienced developers.

Development Standardization

In many organizations, IBM i development practices have evolved organically over decades, leading to inconsistent approaches across teams and projects. This lack of standardization makes it difficult to maintain code quality, share resources between teams, or onboard new developers.

Merlin provides a consistent, integrated set of tools that encourages standardized development practices. It offers built-in code formatting, quality checks, and workflow templates that help teams align on best practices. The integration with Git also encourages standard branching strategies and code review processes that improve overall software quality.

By centralizing development in a container-based solution, Merlin ensures that all developers are using the same tools with the same configurations, eliminating the “works on my machine” problems that often arise from inconsistent development environments.

Cross-Platform Development

Modern applications increasingly span multiple platforms, with IBM i systems often integrated with web, mobile, and cloud services. Traditional IBM i development tools were focused solely on the platform itself, creating silos between development teams working on different components of the same application.

Merlin breaks down these silos by providing a development experience that accommodates cross-platform development. The VS Code-compatible IDE supports multiple languages and platforms, allowing developers to work on IBM i code alongside JavaScript, Python, or other languages used in connected systems.

This unified experience reduces context switching for developers working across platforms and encourages more holistic application development. It enables teams to develop end-to-end solutions that integrate IBM i's robust transaction processing capabilities with modern user experience technologies, creating applications that leverage the best of both worlds.

IBM i Merlin Architecture and Components

IBM i Merlin version 2 delivers a modern development experience through an IBM Certified Container, designed to operate seamlessly within Red Hat OpenShift. This containerized deployment offers significant benefits, including streamlined deployment, enhanced scalability, and consistent performance across diverse environments. OpenShift's flexibility allows for deployment on-premises, on Power servers, or within cloud environments like IBM Cloud (IBM Power Virtual Servers) and other OpenShift-compatible cloud providers. For organizations already utilizing cloud infrastructure, Merlin's integration into their existing OpenShift ecosystem presents a logical and efficient advancement. Furthermore, Merlin supports Single Node OpenShift (SNO), allowing for a reduced footprint and simplified deployment on a single server, as detailed in “Single node OpenShift” on page 285.

This latest iteration of IBM i Merlin signifies a substantial leap forward in IBM i development tools, architected as a complete container-based solution that leverages the robust capabilities of Red Hat OpenShift. Embracing cloud-native principles, Merlin provides exceptional flexibility and scalability. Its microservices-based architecture ensures that each component delivers specialized functionalities while maintaining seamless communication through standardized APIs, fostering a modular and adaptable development environment.

Merlin's architecture, which is shown in Figure 9-16, is strategically designed around the separation of concerns, effectively isolating development activities (IDE), modernization functions (ARCAD tools), and DevOps processes (CI/CD) into distinct, yet interconnected, services. This approach empowers organizations to scale individual components according to their specific requirements without impacting the overall system's stability or performance, optimizing resource utilization and enhancing development agility.

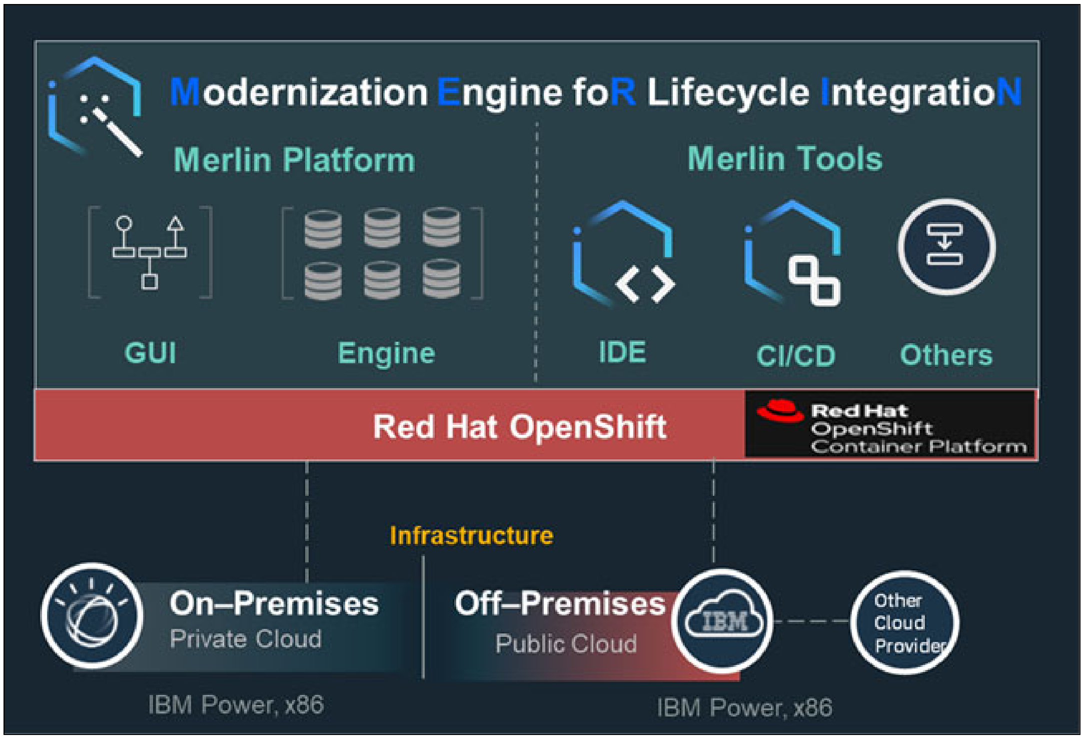


Figure 9-16 Merlin architecture

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 OpenShift container platform provides automated scaling, self-healing capabilities, and consistent deployment across diverse infrastructure environments.

Merlin leverages OpenShift's capabilities for deployment and lifecycle management. This approach ensures that Merlin maintains optimal configuration regardless of the underlying infrastructure. The deployment flexibility allows Merlin to run on Power servers with OpenShift installed directly, in IBM Cloud through Power Virtual Servers, or in any cloud environment that supports OpenShift. This flexibility is particularly valuable for organizations pursuing hybrid cloud strategies, as it enables consistent development experiences across diverse deployment models.

Merlin is deployed as an IBM Certified Container, which ensures compatibility and support across different OpenShift environments. This certification provides assurance that the container meets IBM's standards for security, scalability, and integration capabilities.

DevSpaces Integration

A key architectural component in Merlin version 2 is the integration with OpenShift DevSpaces (formerly CodeReady Workspaces). DevSpaces provides containerized development environments that are defined as code and can be instantiated on demand.

Within Merlin's implementation, DevSpaces creates isolated development environments for each developer or project. These environments contain all necessary tools, dependencies, and access configurations according to organizational standards. When a developer initiates a session, DevSpaces provisions a container with the complete development stack, accessible through a browser interface.

This architecture eliminates “works on my machine” problems by ensuring all developers use identical environments. It also simplifies onboarding by removing the need for complex local setup processes. New team members can become productive within minutes rather than days, as all tools and configurations are provisioned automatically.

IBM Licensing Service

Merlin incorporates the IBM Licensing Service, which provides a unified approach to license management across all Merlin components. This service maintains visibility into license usage, monitors compliance, and simplifies the license renewal process.

The licensing service employs a subscription model that aligns with modern cloud consumption patterns. Organizations pay for actual usage rather than maximum capacity, optimizing license costs while maintaining flexibility to scale as needed. The service includes a dashboard for license administrators to monitor usage patterns and allocation efficiency.

IBM Certificate Manager

Security is fundamental to Merlin's architecture, and the IBM Certificate Manager plays a role in the security framework. This component handles certificate lifecycle management, including issuance, renewal, and revocation processes.

The Certificate Manager integrates with certificate authorities while providing options for development environments. It enforces certificate policies and implements rotation schedules for security compliance.

By centralizing certificate management, Merlin reduces the security risks associated with expired certificates or improper implementations. The implementation includes certificate monitoring that alerts administrators about upcoming expirations.

Browser-Based IDE

The centerpiece of Merlin's development experience is its browser-based IDE, built on Visual Studio Code. The IDE features code intelligence for IBM i languages, including RPG, CL, DDS, and SQL. It provides syntax checking, code completion, and documentation that accelerates the development process. The editor supports split views, allowing developers to see related files side by side, such as a display file and its associated RPG program.

Debugging capabilities in version 2 include support for breakpoints, watch expressions, and call stack visualization. The IDE includes integrated terminal access with IBM i command support, enabling developers to execute commands directly from the development environment without switching contexts.

CI/CD Toolchain

Merlin delivers a CI/CD toolchain centered around Jenkins with IBM i-specific capabilities. 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 like ILE service programs. It integrates with source control to trigger builds automatically when code changes are detected, ensuring that the latest code is always built and tested.

Deployment automation handles the complexity of moving objects between IBM i environments while maintaining dependencies and ensuring proper sequencing. The implementation supports deployment strategies that reduce risk during updates to critical systems.

The pipeline visualization tools provide status updates and metrics, helping teams identify bottlenecks and improve their delivery processes. These visualizations include stage information, timing data, and failure analytics that help identify the root causes of pipeline issues.

ARCAD Tools Integration

Merlin version 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.

- ARCAD Transformer

  The Transformer component provides automated conversion from fixed-format RPG to free-format modern RPG. This tool goes beyond simple syntax transformation by applying modern programming patterns during the conversion process.

  The implementation includes transformations that consider the structure of the code rather than just its syntax. For example, it can recognize common programming patterns and transform them into more maintainable modern structures. The tool also identifies potential issues like improper indicator usage or outdated file access methods.

  The analysis capabilities allow developers to compare original and transformed code side by side, with explanations of transformations. This transparency builds developer confidence in the automated transformations and serves as an educational tool for learning modern RPG techniques.

- ARCAD Observer

  Observer provides application analysis and visualization capabilities that help developers understand complex applications. Observer includes analysis that identifies patterns and relationships within applications.

The application mapping features create visual representations of object dependencies, call hierarchies, and data flows. These maps help developers understand the impact of proposed changes before they're implemented, reducing the risk associated with modifying complex systems. The implementation includes visualization capabilities that allow developers to explore relationships, focusing on areas of interest.

The code quality analysis evaluates applications against industry best practices and organizational standards. It identifies potential issues like unreachable code, inefficient algorithms, or security vulnerabilities. The analysis provides recommendations for remediation, prioritized by potential impact.

Observer's documentation generation capabilities create comprehensive documentation that reflects the current state of applications. This feature is particularly valuable for legacy applications where original documentation may be outdated or missing entirely. The documentation includes both technical details for developers and business-oriented explanations for stakeholders.

- ARCAD Builder

  Builder manages the build processes required for IBM i applications, ensuring consistent, reproducible builds across environments. It includes support for build optimization that improves build times for applications.

  The dependency management features determine the correct build sequence based on object relationships, reducing the need for manually maintained build scripts. The dependency analysis includes direct references and dependencies like data areas or shared files.

  The incremental build capabilities optimize the build process by rebuilding only objects affected by changes. This approach reduces build times for large applications where changes typically affect only a portion of the codebase.

  Builder integrates with the CI/CD pipeline to provide build metrics and status information. The implementation includes analytics that can identify build patterns and suggest optimizations to improve build efficiency.

Integration and Workflow

What distinguishes Merlin is the integration between these components. The browser-based IDE connects to source control systems, triggering automated builds and deployments when code is committed. The ARCAD tools provide insights within the development environment, helping developers make informed decisions without switching contexts. Figure 9-17 shows how the development components fit together in a Merlin environment.

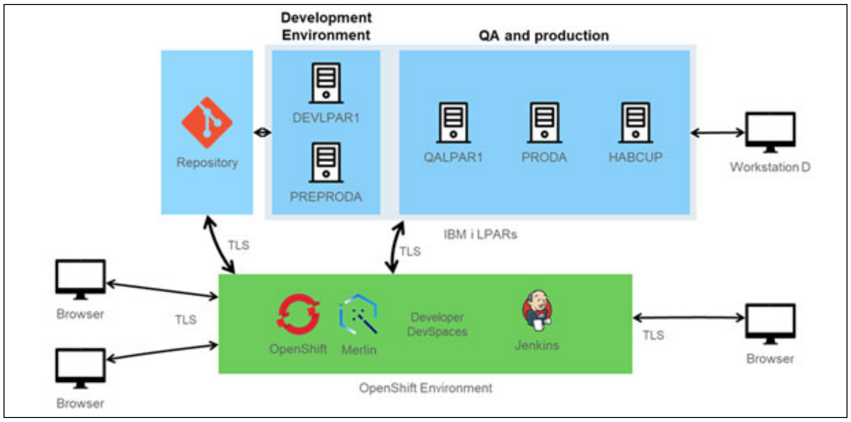


Figure 9-17 Merlin development topology

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, allowing developers to focus on creating business value rather than managing development infrastructure.

Single node OpenShift

Red Hat OpenShift's standard architecture, designed for high availability, mandates a minimum of three control plane nodes and three worker nodes. This requirement translates to at least six logical partitions (LPARs) when deploying on an IBM Power server, a significant hurdle for many clients. However, Red Hat's introduction of Single Node OpenShift (SNO) offers a more streamlined approach, allowing for the deployment of an OpenShift cluster within a single LPAR on IBM Power.

Given the inherent high availability of IBM Power LPARs, the added redundancy provided by a traditional multi-node OpenShift cluster becomes less critical. This is particularly relevant when running an OpenShift cluster alongside an IBM i partition for a Merlin environment. Utilizing SNO in this scenario provides clients with greater flexibility in their Merlin setup, optimizing resource usage and simplifying deployment without sacrificing essential reliability within the robust IBM Power ecosystem.

Single Node OpenShift (SNO) represents a specialized configuration of the robust OpenShift container platform, designed for scenarios where a full-scale, highly available cluster might be excessive. In essence, SNO consolidates the control plane and worker node functionalities onto a single server. This streamlined architecture proves advantageous in resource-constrained environments, edge computing deployments, and for development or testing purposes. While it sacrifices the high availability inherent in multi-node clusters, SNO provides a practical means of leveraging OpenShift's capabilities in situations demanding a smaller footprint.

The practicality of SNO stems from its ability to deliver the core benefits of OpenShift, such as container orchestration and application management, in a simplified and more accessible manner. This makes it particularly valuable for use cases like remote locations with limited infrastructure, or for developers seeking a rapid and efficient way to experiment with containerized applications. However, it is important to remember that because it is a single node, that if that node where to fail, the entire OpenShift instance would fail. So it is not recommended for production workloads that require constant up time.

For more details on Merlin including installation instructions and a user guide, refer to IBM Redbooks publication *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 within the IBM i operating system. This tight integration is a key characteristic, offering users a simplified and streamlined database management experience. Unlike other database systems that require separate installation and configuration, DB2 for IBM i is inherently part of the operating system, which contributes to its ease of use and management. It provides a robust and reliable platform for handling diverse application needs, from traditional host-based systems to modern client/server and business intelligence applications.

A significant strength of DB2 for IBM i lies in its comprehensive feature set, including triggers, stored procedures, and dynamic bitmapped indexing. These capabilities enable developers to build sophisticated and efficient database applications. Furthermore, its adherence to industry-standard SQL ensures compatibility and flexibility in data manipulation and querying. The system's ability to seamlessly work with both IBM i files and SQL tables further enhances its versatility. Moreover, tools like the IBM DB2 Query Manager and SQL Development Kit for i provide developers with the necessary resources to effectively interact with and leverage the power of DB2 for IBM i.

9.3.1 Modernization Techniques for IBM Db2 for i

Over the years, IBM Db2 for i has evolved from a traditional relational database into a powerful ecosystem that supports both structured and unstructured data, advanced analytics, and procedural programming. This chapter explores key features of Db2 for i that can supercharge your SQL capabilities, enabling efficient data management, analysis, and performance optimization.

The complete example for this section can be found in this repo:
https://github.com/NielsLiisberg/sql-on-steroids

The catalog of services used in this section can be found by ruining this SQL query:

```
select * from qsys2.services_info;
```

Building a Sample Schema

Before diving into advanced SQL techniques, we need a dataset to work with. Creating a sample schema in Db2 for i is straightforward:

```
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 (CTEs)

Common Table Expressions (CTEs) allow for more readable and maintainable queries, especially when dealing with complex joins or recursive structures. Example 9-1 is an example that constructs full employee names for the query.

Example 9-1 Constructing full employee name

```
with emp_full_name as (
    select rtrim(firstnme) || ' ' || midinit || ' ' || lastname as full_name, *
    from sqlxxl.employee
)
select * from emp_full_name
where full_name like '%JOHN%';
```

Recursive SQL

Recursive queries help in hierarchical data traversal, such as organizational structures. Example 9-2 shows a query which retrieves department hierarchies – note the **“connect by”.**

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 sqlxxl.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 sqlxxl.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 use of **`REGEXP_REPLACE`**.

Example 9-4 Regular expression support

```
values regexp_replace('AMOUNT: 123,456.78$', '[^0-9.]', '');
```

This removes all non-numeric characters except for the decimal point.

Geospatial Functions

Db2 for i includes geospatial capabilities for location-based data. Example 9-5 adds latitude and longitude points to an organization table.

Example 9-5 Geospatial support

```
alter table sqlxxl.org add column location_point qsys2.st_point;
update sqlxxl.org
set location_point = qsys2.st_point('POINT(-74.1836 40.7261)')
where deptnumb = '10';
```

Distance calculations and mapping integrations are also possible using built-in functions.

Compound Statements

Db2 for i allows multiple SQL statements to be grouped using compound statements as shown in Example 9-6.

Example 9-6 Example of compound statement

```
begin
    for
        select *
        from   qsys2.output_queue_entries_basic
        where  output_queue_name = 'QEZJOBLOG'
        and    create_timestamp < now() - 7 days
    do
```

```
        call qcmdexc('DLTSPLF FILE(' || SPOOLED_FILE_NAME
            || ') JOB(' || JOB_NAME || ') SPLNBR(' || FILE_NUMBER || ')');
    end for;
end;
```

This groups multiple statements in a structured execution block.

Stored Procedures

Stored procedures enable reusable, parameterized SQL logic. Example 9-7 shows a stored procedure for deleting joblog.

Example 9-7 Example using stored procedures

```
create or replace procedure sqlxxl.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 = 'QEZJOBLOG'
        and    create_timestamp < now() - days_to_keep days
    do
        call qcmdexc('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

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 = 'QEZJOBLOG'
        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 a 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 a http request.

Example 9-11 XML processing

```
select *
from xmltable(
    '/exchangerates/dailyrates/currency'
    passing xmlparse (
      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 INSTEAD OF trigger

```
create or replace view sqlxxl.emp_full_name as (
    select empno,
           rtrim(firstnme) concat ' ' concat midinit concat ' ' concat lastname as
full_name
    from sqlxxl.employee
);
create or replace trigger sqlxxl.emp_full_name
instead of update on sqlxxl.emp_full_name
referencing new as new_row old as old_row
for each row mode db2row
begin
   if updating then
       update sqlxxl.employee
       set firstnme = 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 sqlxxl.departmnt (
    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 sqlxxl.departmnt_hist like sqlxxl.departmnt;
alter table sqlxxl.departmnt add versioning use history table
sqlxxl.department_hist;
select * from sqlxxl.departmnt for system_time as of current timestamp - 6 months;
```

9.3.2 Summary

The advanced SQL features within IBM Db2 for i, including recursive queries, OLAP functions, geospatial processing, triggers, and temporal tables, underscore its position as a leading-edge database platform. Mastery of these techniques allows for the implementation of powerful data management and analytical workflows. As Db2 for i evolves, these skills become essential for constructing contemporary, reliable, and efficient database architectures.

9.3.3 References

Here is some additional documentation on the topics covered in this section.

- All examples
  - https://github.com/NielsLiisberg/sql-on-steroids
- Cte
  - https://www.ibm.com/docs/en/i/7.5?topic=queries-using-recursive-common-table-expressions-recursive-views
- OLAP
  - https://www.ibm.com/docs/en/i/7.5?topic=statement-using-olap-specifications
- Regex
  - https://www.ibm.com/docs/en/i/7.5?topic=queries-using-recursive-common-table-expressions-recursive-views
- Geospatial
  - https://www.ibm.com/docs/en/i/7.5?topic=analytics-geospatial-functions
- Compound
  - https://www.ibm.com/docs/en/i/7.3?topic=pl-compound-statement
- UDF
  - https://www.ibm.com/docs/en/i/7.5?topic=statements-create-function-sql-scalar
- UDTF
  - https://www.ibm.com/docs/en/i/7.5?topic=statements-create-function-sql-table
- JSON
  - https://www.ibm.com/docs/en/i/7.5?topic=data-using-json-table
- XM
  - https://www.ibm.com/docs/en/i/7.5?topic=programming-sql-statements-sqlxml-functions
- Triggers
  - https://www.ibm.com/docs/en/i/7.5?topic=statements-create-trigger

9.4 Open-source Databases on IBM i

This section covers open-source database that can be used on any modernization project on the IBM i server. IBM i, renowned for its robust and secure environment, has embraced open-source technologies to offer a wider range of database solutions.

This section explores three currently available and popular open-source databases that can be effectively deployed on IBM i: MariaDB, PostgreSQL and SQLite.

MariaDB

MariaDB is a community-driven version of MySQL. MariaDB maintains strong compatibility while incorporating innovative features and performance enhancements.

The key features of MariaDB are:

- High Performance

  MariaDB is known for its speed and efficiency, making it suitable for demanding workloads.
- Robustness:

  MariaDB offers a stable and reliable platform with a proven track record.
- Active Community

MariaDB benefits from a large and active community – providing support, resources and ongoing development.

MariaDB is ideal for a wide range of applications, including web applications, e-commerce platforms and data warehousing.

PostgreSQL

Overview: A powerful heavyweight and feature-rich relational database system, which is known for its advanced features and strong emphasis on standards compliance.

The key Features of PostgreSQL are:

- 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 well-suited for complex applications, data analysis, and demanding workloads requiring advanced features.

SQLite

SQLite is a lightweight and embedded SQL database engine known for its simplicity and ease of use. It is a great candidate for quickly and simply storing data.

The key features of SQLite are:

- Lightweight

  SQLite is compact and easy to embed in applications.
- File-Based

  Stores data in a single file, making it easy to distribute and manage, especially when data-replication is a requirement.
- No Server Required

  Eliminates the need for a separate database server, reducing complexity.

SQLite is ideal for mobile and embedded applications, small-scale projects and situations where a lightweight database is required.

Choosing the Right Database

This section has provided you with a quick introduction into the databases available on IBM i. More details are provided in the following sections.

While DB2 for i is the main database, and quite rightly so, there are other options out there with these open-source offerings that should be considered when choosing a database for your application. The best choice of database depends on your specific needs and requirements, and you need to consider factors such as:

Workload The type and volume of data you will be handling.

Features The specific features required for your application (e.g., full-text search, spatial data support).

Performance The required performance and scalability of the database.

Ease of Use The level of technical expertise required to manage and maintain the database.

Table 9-1 provides a high level comparison of the database options in several areas. Scores range from 1 to 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

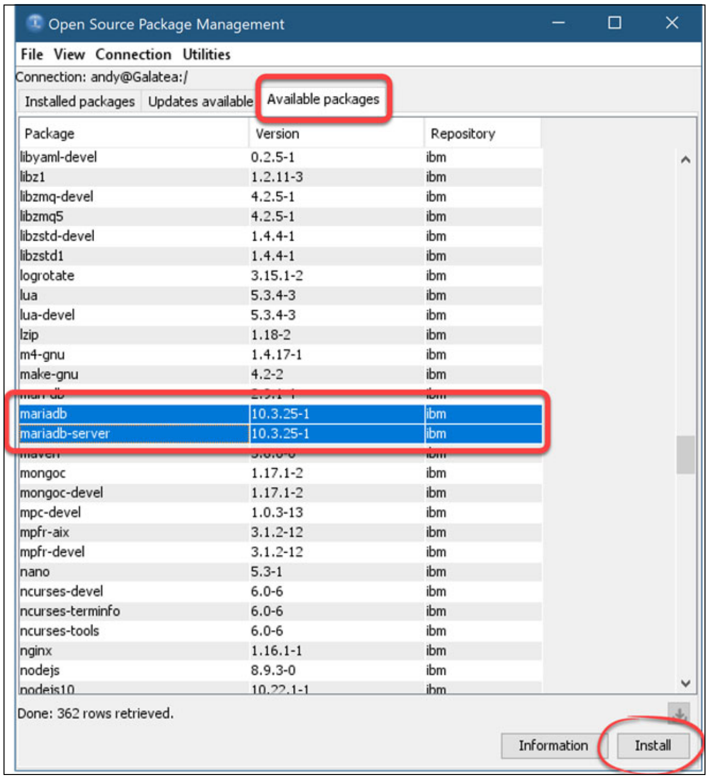
9.4.1 MariaDB

Maria DB is a replacement for MySQL. It is open source and is developed by the original MySQL developers. MariaDB is one of the most popular database servers in the world. MariaDB is often chosen because it is fast, scalable and robust.

MariaDB is a community developed branch of MySQL. When using MariaDB you will notice many of the commands use the MySQL naming convention.

Installing MariaDB

The easiest way to install MariaDB is to use IBM Access for Client Solutions (ACS) open-source package management and look for the two MariaDB packages and install them as shown in Figure 9-18.



Connection: andy@Galatea:/

Package	Version	Repository
libyaml-devel	0.2.5-1	ibm
libz1	1.2.11-3	ibm
libzmq-devel	4.2.5-1	ibm
libzmq5	4.2.5-1	ibm
libzstd-devel	1.4.4-1	ibm
libzstd1	1.4.4-1	ibm
logrotate	3.15.1-2	ibm
lua	5.3.4-3	ibm
lua-devel	5.3.4-3	ibm
lzip	1.18-2	ibm
m4-gnu	1.4.17-1	ibm
make-gnu	4.2-2	ibm
man-db	2.9.1-1	ibm
mariadb	10.3.25-1	ibm
mariadb-server	10.3.25-1	ibm
maven	[illegible]	ibm
mongoc	1.17.1-2	ibm
mongoc-devel	1.17.1-2	ibm
mpc-devel	1.0.3-13	ibm
mpfr-aix	3.1.2-12	ibm
mpfr-devel	3.1.2-12	ibm
nano	5.3-1	ibm
ncurses-devel	6.0-6	ibm
ncurses-terminfo	6.0-6	ibm
ncurses-tools	6.0-6	ibm
nginx	1.16.1-1	ibm
nodejs	8.9.3-0	ibm
nodejs10	10.22.1-1	ibm

Done: 362 rows retrieved.

Figure 9-18 Package list

You can also install MariaDB using Yum. To install the packages we need to open a Shell session, with QSECOFR authority.

To use yum, run the command below, in a shell session

```
yum install mariadb mariadb-server
```

Configuring MariaDB

The next step is to configure MariaDB. Follow the following steps:

1. In a shell session, run the MariaDB setup procedure. This is performed by running the command below.

   ```
   mysql_install_db --user=mysql
   ```

2. Once this is complete, the next step is to set the root password using the following command.

   ```
   mysqladmin -u root password ******
   ```

3. Next, we need to issue the `GRANT` command to apply the appropriate permissions. In a shell session we can use the following command.

   ```
   mysql -u root -p
   ```

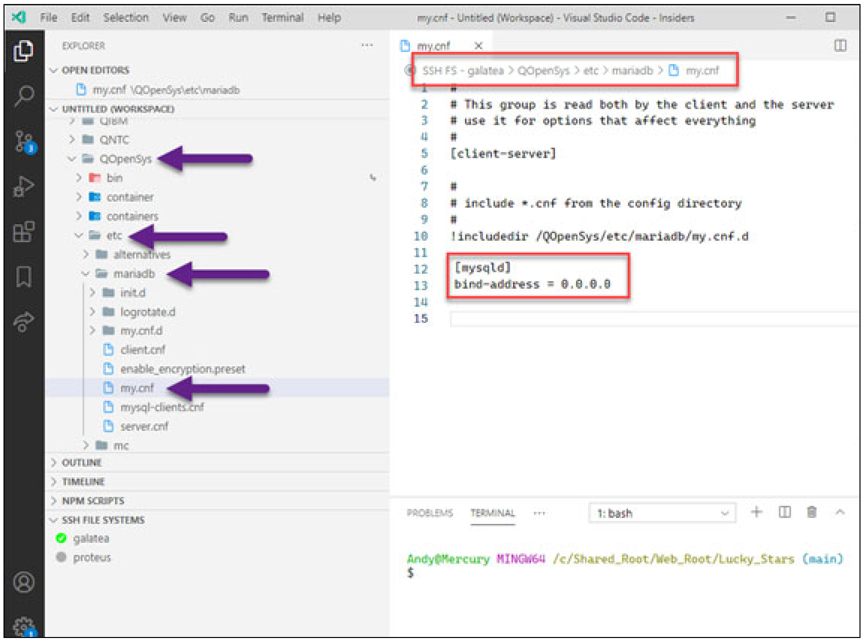
   Then we can grant root authority to the root userid.

   ```
   GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'your password'
   ```

   With that command, we have told MariaDB to Grant the Privileges of type ALL to root. These privileges are for all databases and it applies to all tables of that database, which is indicated by the **.**.

   These privileges are assigned to username root when that username is connected through any IP addresses (specified by *@%*). *IDENTIFIED BY* secures it using your password.

4. Before starting the database server, we need to ensure it is running on the correct IP address and port number. The file *my.cnf* holds the configuration. This can be seen in Figure 9-19.



```
#
# This group is read both by the client and the server
# use it for options that affect everything
#
[client-server]

#
# include *.cnf from the config directory
#
!includedir /QOpenSys/etc/mariadb/my.cnf.d

[mysqld]
bind-address = 0.0.0.0

```

```
Andy@Mercury MINGW64 /c/Shared_Root/Web_Root/Lucky_Stars (main)
$
```

Figure 9-19 MariaDB configuration

Here I have added to listen on all our IP addresses. By default, it uses port 3306. If that port is in use by your corporate network, it can be changed here.

Starting Maria DB Server

To start the MariaDB server run the following command from a shell session:

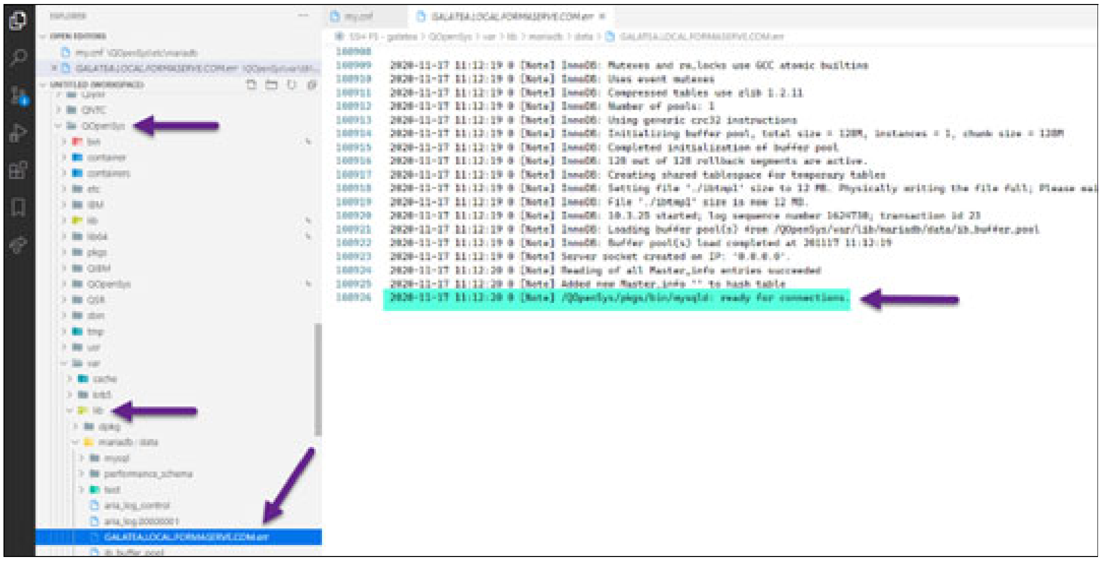
```
mysqld_safe --datadir=/QOpenSys/var/lib/mariadb/data
```

When this command has been run, it will tie up the terminal and if the terminal is closed it will stop the database server from running. This option is acceptable for initial testing.

Once we are happy with testing our database it would be better to submit the starting of the server as a batch job. This command for this would be:

```
SBMJOB CMD(QSH CMD('/QOpenSys/pkgs/bin/mysqld_safe
--datadir=/QOpenSys/var/lib/mariadb/data')) JOB(STR_MARIA)
```

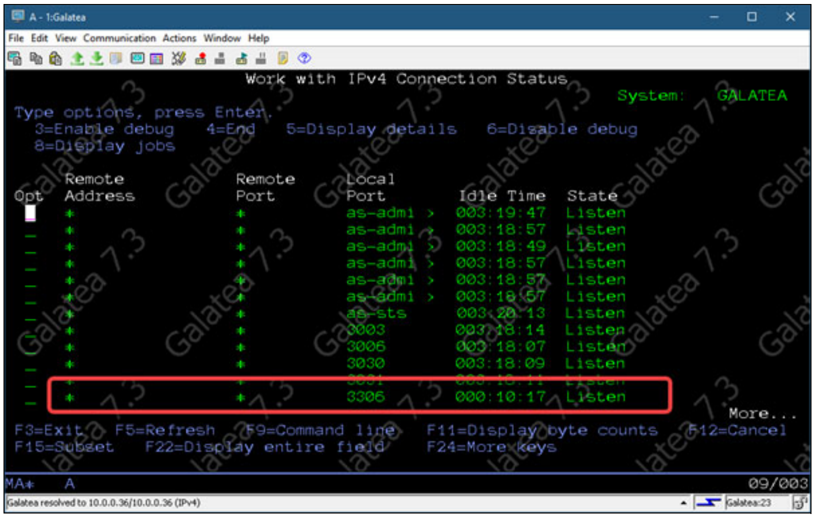
If we check the error log, shown in Figure 9-20, we can see the server is ready and waiting for connections.



```
2020-11-17 11:12:19 0 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
2020-11-17 11:12:19 0 [Note] InnoDB: Uses event mutexes
2020-11-17 11:12:19 0 [Note] InnoDB: Compressed tables use zlib 1.2.11
2020-11-17 11:12:19 0 [Note] InnoDB: Number of pools: 1
2020-11-17 11:12:19 0 [Note] InnoDB: Using generic crc32 instructions
2020-11-17 11:12:19 0 [Note] InnoDB: Initializing buffer pool, total size = 128M, instances = 1, chunk size = 128M
2020-11-17 11:12:19 0 [Note] InnoDB: Completed initialization of buffer pool
2020-11-17 11:12:19 0 [Note] InnoDB: 128 out of 128 rollback segments are active.
2020-11-17 11:12:19 0 [Note] InnoDB: Creating shared tablespace for temporary tables
2020-11-17 11:12:19 0 [Note] InnoDB: Setting file './ibtmp1' size to 12 MB. Physically writing the file full; Please wai
2020-11-17 11:12:19 0 [Note] InnoDB: File './ibtmp1' size is now 12 MB.
2020-11-17 11:12:19 0 [Note] InnoDB: 10.3.25 started; log sequence number 1624730; transaction id 23
2020-11-17 11:12:19 0 [Note] InnoDB: Loading buffer pool(s) from /QOpenSys/var/lib/mariadb/data/ib_buffer_pool
2020-11-17 11:12:19 0 [Note] InnoDB: Buffer pool(s) load completed at 201117 11:12:19
2020-11-17 11:12:19 0 [Note] Server socket created on IP: '0.0.0.0'.
2020-11-17 11:12:20 0 [Note] Reading of all Master_info entries succeeded
2020-11-17 11:12:20 0 [Note] Added new Master_info '' to hash table
2020-11-17 11:12:20 0 [Note] /QOpenSys/pkgs/bin/mysqld: ready for connections.
```

Figure 9-20 Validate server is ready

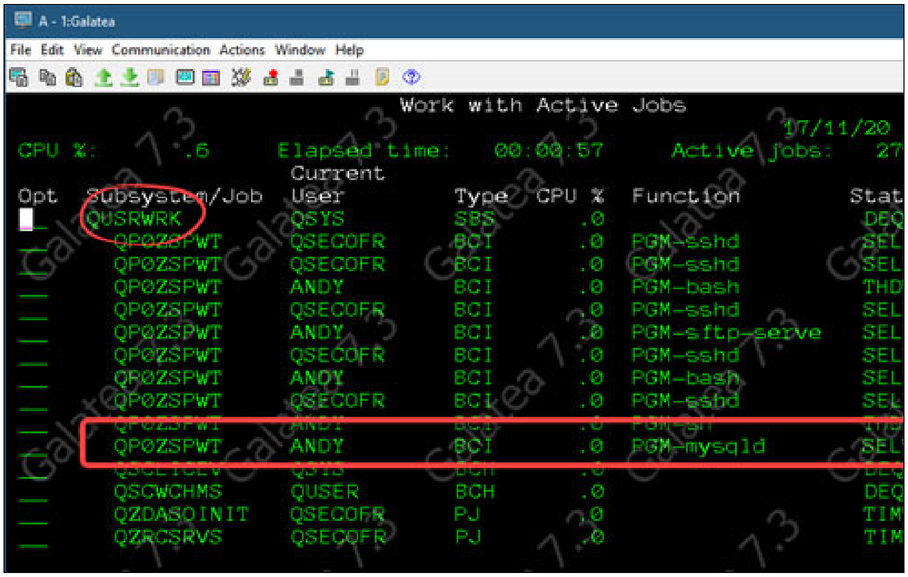
If you are using a 5250 terminal, you can run the **NETSTAT *CNN** command, and look for port *3306 as shown in Figure 9-21*.



```
                     Work with IPv4 Connection Status
                                                        System:   GALATEA
Type options, press Enter.
  3=Enable debug   4=End   5=Display details   6=Disable debug
  8=Display jobs

     Remote           Remote     Local
Opt  Address          Port       Port          Idle Time   State
_    *                *          as-admi >     003:19:47   Listen
_    *                *          as-admi >     003:18:57   Listen
_    *                *          as-admi >     003:18:49   Listen
_    *                *          as-admi >     003:18:57   Listen
_    *                *          as-admi >     003:18:57   Listen
_    *                *          as-admi >     003:18:57   Listen
_    *                *          as-sts        003:20:13   Listen
_    *                *          3003          003:18:14   Listen
_    *                *          3006          003:18:07   Listen
_    *                *          3030          003:18:09   Listen
_    *                *          3001          003:18:11   Listen
_    *                *          3306          000:10:17   Listen
                                                                    More...
F3=Exit   F5=Refresh   F9=Command line   F11=Display byte counts   F12=Cancel
F15=Subset   F22=Display entire field    F24=More keys

MA*    A                                                            09/003
Galatea resolved to 10.0.0.36/10.0.0.36 (IPv4)                 Galatea:23
```

Figure 9-21 NETSTAT command

Or look for the job in **WRKACTJOB**. The server job will run by default in the *QUSRWRK* subsystem as shown in Figure 9-22.



```
                                 Work with Active Jobs
                                                                    17/11/20
 CPU %:       .6      Elapsed time:   00:00:57     Active jobs:   27
                          Current
 Opt  Subsystem/Job   User          Type  CPU %  Function            Stat
      QUSRWRK         QSYS          SBS      .0                       DEQ
 __     QP0ZSPWT      QSECOFR       BCI      .0  PGM-sshd             SEL
 __     QP0ZSPWT      QSECOFR       BCI      .0  PGM-sshd             SEL
 __     QP0ZSPWT      ANDY          BCI      .0  PGM-bash             THD
 __     QP0ZSPWT      QSECOFR       BCI      .0  PGM-sshd             SEL
 __     QP0ZSPWT      ANDY          BCI      .0  PGM-sftp-serve       SEL
 __     QP0ZSPWT      QSECOFR       BCI      .0  PGM-sshd             SEL
 __     QP0ZSPWT      ANDY          BCI      .0  PGM-bash             SEL
 __     QP0ZSPWT      QSECOFR       BCI      .0  PGM-sshd             SEL
 __     QP0ZSPWT      ANDY          BCI      .0  PGM-sh               THD
 __     QP0ZSPWT      ANDY          BCI      .0  PGM-mysqld           SEL
 __     QSCLICEV      QSYS          BCH      .0                       DEQ
 __     QSCWCHMS      QUSER         BCH      .0                       DEQ
 __     QZDASOINIT    QSECOFR       PJ       .0                       TIM
 __     QZRCSRVS      QSECOFR       PJ       .0                       TIM
```

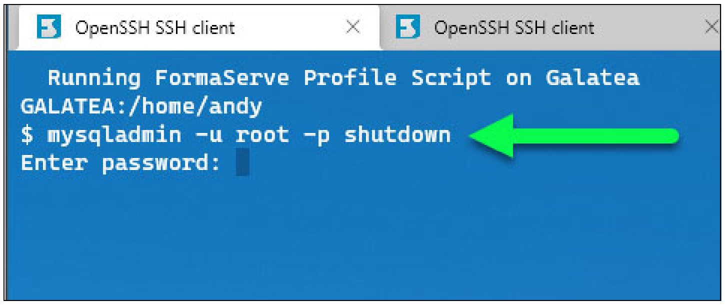
Figure 9-22   Using WRKACTJOB screen

Stopping the Server

If you need to stop the database server for any reason, run the following command in a shell session.

```
mysqladmin -u root -p shutdown
```

This command will prompt for the root password, then shutdown the server job. This is 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

Using MariaDB

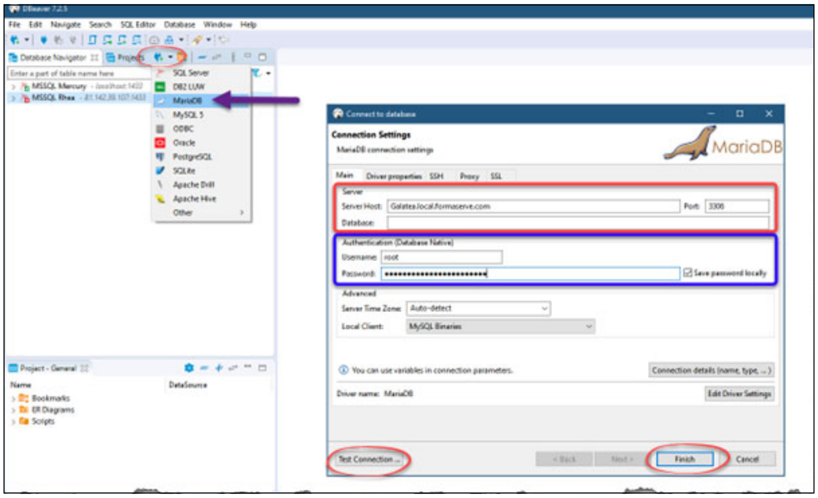
Now we have MariaDB installed, how do we go about using this database?

1. Firstly, we need a database client to access it. There are loads of options out there, Google is your friend for once.

   Here we will be using DBeaver, an open-source database client that can be used for many databases, not only MariaDB. DBeaver can be downloaded from `https://dbeaver.io/download/`.

2. When we first open the beaver inteface, we need to create a database connection to our IBM i.

   This can be achieved as shown in Figure 9-24.



Server

Server Host: Galatea.local.formaserve.com Port: 3306

Database:

Authentication (Database Native)

Username: root

Password: •••••••••••••••••••••• ☑ Save password locally

Advanced

Server Time Zone: Auto-detect

Local Client: MySQL Binaries

You can use variables in connection parameters.

Driver name: MariaDB

Figure 9-24 Create database connection

3. We take the new connection icon, then select the MariaDB type and fill the connection settings for both the server and user details.

   And now we can see our MariaDB database on our IBM i as shown in Figure 9-25.

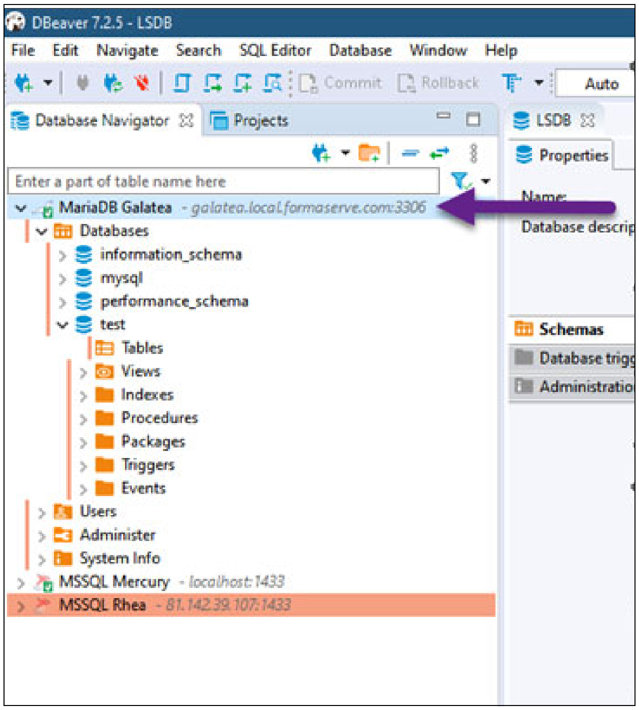


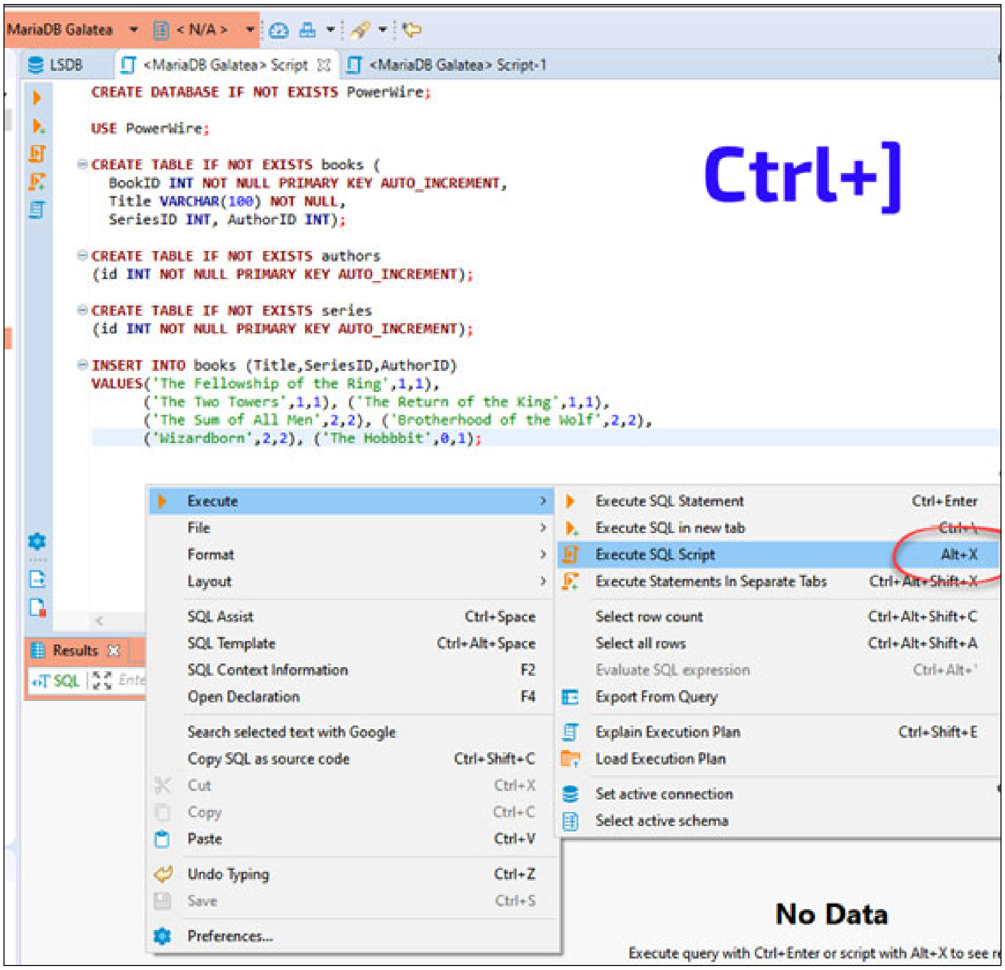
Figure 9-25 Seeing the connection

Creating a Database and Table

Now we have our database server running and a client that connects to it. Next create a database and create a couple of tables within it.

In our example we download a SQL script from the MariaDB site that will create these database objects for us.

1. Use *Ctrl+]* to open a new SQL window, then *Alt+X* to execute the script as shown in Figure 9-26.

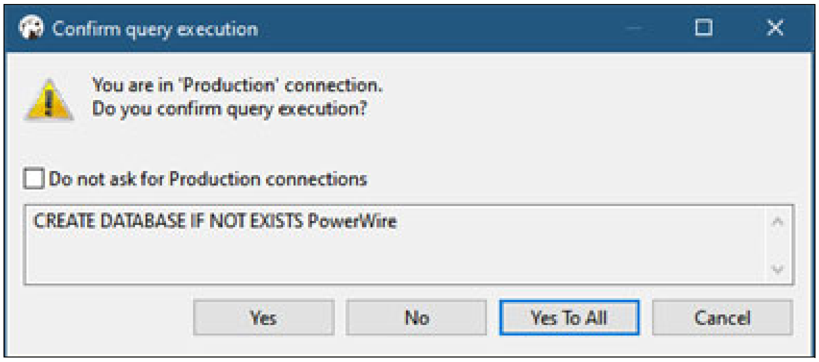


```
CREATE DATABASE IF NOT EXISTS PowerWire;

USE PowerWire;

CREATE TABLE IF NOT EXISTS books (
  BookID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  Title VARCHAR(100) NOT NULL,
  SeriesID INT, AuthorID INT);

CREATE TABLE IF NOT EXISTS authors
(id INT NOT NULL PRIMARY KEY AUTO_INCREMENT);

CREATE TABLE IF NOT EXISTS series
(id INT NOT NULL PRIMARY KEY AUTO_INCREMENT);

INSERT INTO books (Title,SeriesID,AuthorID)
VALUES('The Fellowship of the Ring',1,1),
      ('The Two Towers',1,1), ('The Return of the King',1,1),
      ('The Sum of All Men',2,2), ('Brotherhood of the Wolf',2,2),
      ('Wizardborn',2,2), ('The Hobbbit',0,1);
```

MariaDB Galatea
< N/A >
LSDB
<MariaDB Galatea> Script
<MariaDB Galatea> Script-1
Ctrl+]
Execute
File
Format
Layout
SQL Assist Ctrl+Space
SQL Template Ctrl+Alt+Space
SQL Context Information F2
Open Declaration F4
Search selected text with Google
Copy SQL as source code Ctrl+Shift+C
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Undo Typing Ctrl+Z
Save Ctrl+S
Preferences...
Execute SQL Statement Ctrl+Enter
Execute SQL in new tab Ctrl+\
Execute SQL Script Alt+X
Execute Statements In Separate Tabs Ctrl+Alt+Shift+X
Select row count Ctrl+Alt+Shift+C
Select all rows Ctrl+Alt+Shift+A
Evaluate SQL expression Ctrl+Alt+'
Export From Query
Explain Execution Plan Ctrl+Shift+E
Load Execution Plan
Set active connection
Select active schema
Results
SQL
No Data
Execute query with Ctrl+Enter or script with Alt+X to see r

Figure 9-26 Execute script from SQL window

Before it runs, there is a pop-up warning that the server properties state this is a production server and do we want to continue as shown in Figure 9-27.Use the connection properties to change this if you need to.



Confirm query execution

You are in 'Production' connection.
Do you confirm query execution?

☐ Do not ask for Production connections

```
CREATE DATABASE IF NOT EXISTS PowerWire
```

Figure 9-27 Set to not ask for Production connections

2. Now the SQL script has run, we can see all the objects in Figure 9-28.

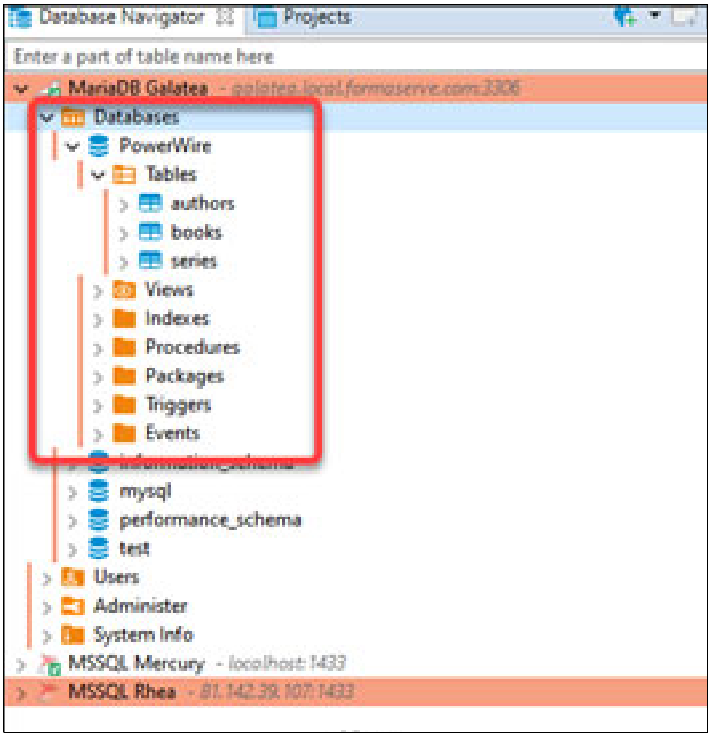
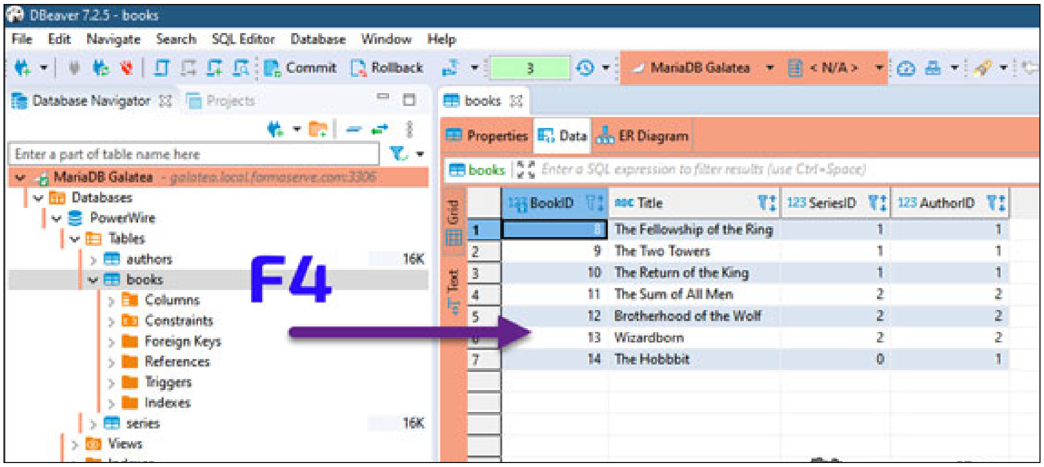


Figure 9-28

The new database and three tables now exist.

3. To see if the inserts worked, I would highlight the books table and take **F4** to view as shown in Figure 9-29.



	BookID	Title	SeriesID	AuthorID
1	8	The Fellowship of the Ring	1	1
2	9	The Two Towers	1	1
3	10	The Return of the King	1	1
4	11	The Sum of All Men	2	2
5	12	Brotherhood of the Wolf	2	2
6	13	Wizardborn	2	2
7	14	The Hobbbit	0	1

Figure 9-29 Validate that the inserts worked

9.4.2 PostgreSQL on IBM i

In this section, we will show you how we can install and use the very popular PostgreSQL database on IBM i. PostgreSQL is an open source relational database management system, a DBMS developed by a worldwide team of developers. It has been around since 1986 and is available on the IBM i.

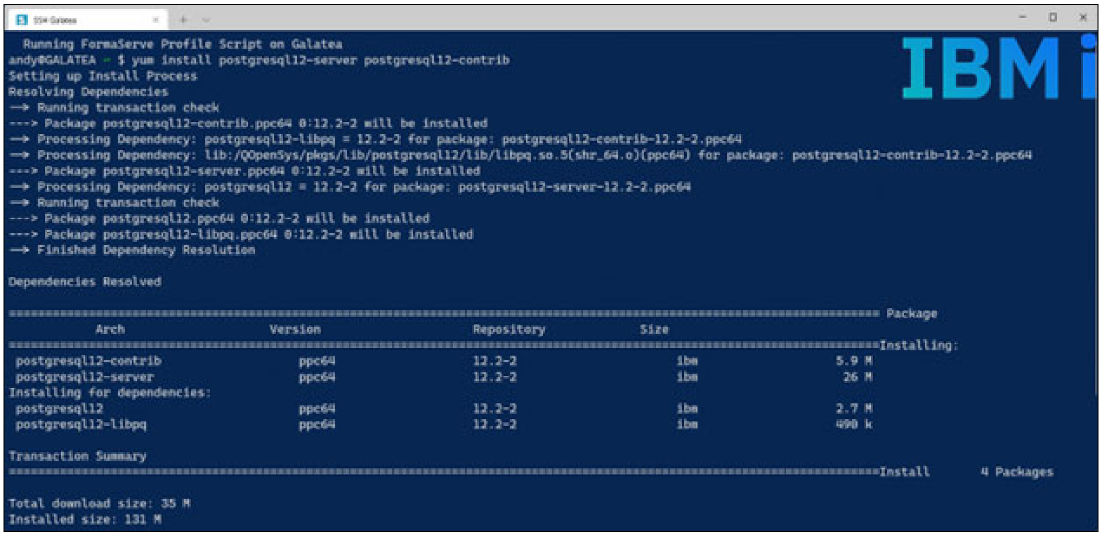
Install

To install this database, we use either IBM Access for Client Solutions (ACS) then Open-Source Package Management, or yum.

To use yum we install PostgreSQL in a shell window using the command below:

```
yum install postgresql12-server postgresql12-contrib
```

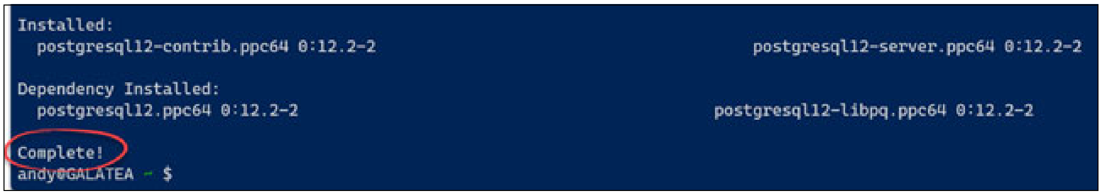
The result is shown in Figure 9-30.



```
 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/pkgs/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

======================================================================================================================= Package
            Arch                     Version                    Repository            Size
=======================================================================================================================Installing:
 postgresql12-contrib                   ppc64                    12.2-2                        ibm                    5.9 M
 postgresql12-server                    ppc64                    12.2-2                        ibm                     26 M
Installing for dependencies:
 postgresql12                           ppc64                    12.2-2                        ibm                    2.7 M
 postgresql12-libpq                     ppc64                    12.2-2                        ibm                    490 k

Transaction Summary
=======================================================================================================================Install       4 Packages

Total download size: 35 M
Installed size: 131 M
```

Figure 9-30   Installing with yum

Look out for the completion message shown in Figure 9-31.



```
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

That’s how easy it is to install this popular open source database.

Setup

Now we have the product installed, we need to do some setup.

1. First we need to create a user to do the setup. We create a user called *POSTGRES*, with *QSECOFR* authority to accomplish this, and additionally specify a home directory of */HOME/POSTGRES*.

   ```
   CRTUSRPRF USRPRF(POSTGRES)
             USRCLS(*SECOFR)
             TEXT('Profile for POSTGRESQL database')
             HOMEDIR('/home/postgres')
   ```

2. Next we have to create the home directory and ensure the directories owner is correct.

   Use the following commands, in a shell window, to achieve this.

```
mkdir /home/postgres
chown postgres /home/postgres
```

In a shell window, sign on using the *POSTGRES* profile and run the following commands:

```
export PGDATA=/home/postgres
export PATH=$PATH:/QOpenSys/pkgs/bin/
```

This will setup our path for all open source packages and let Postgres know where its databases, and environment, are stored.

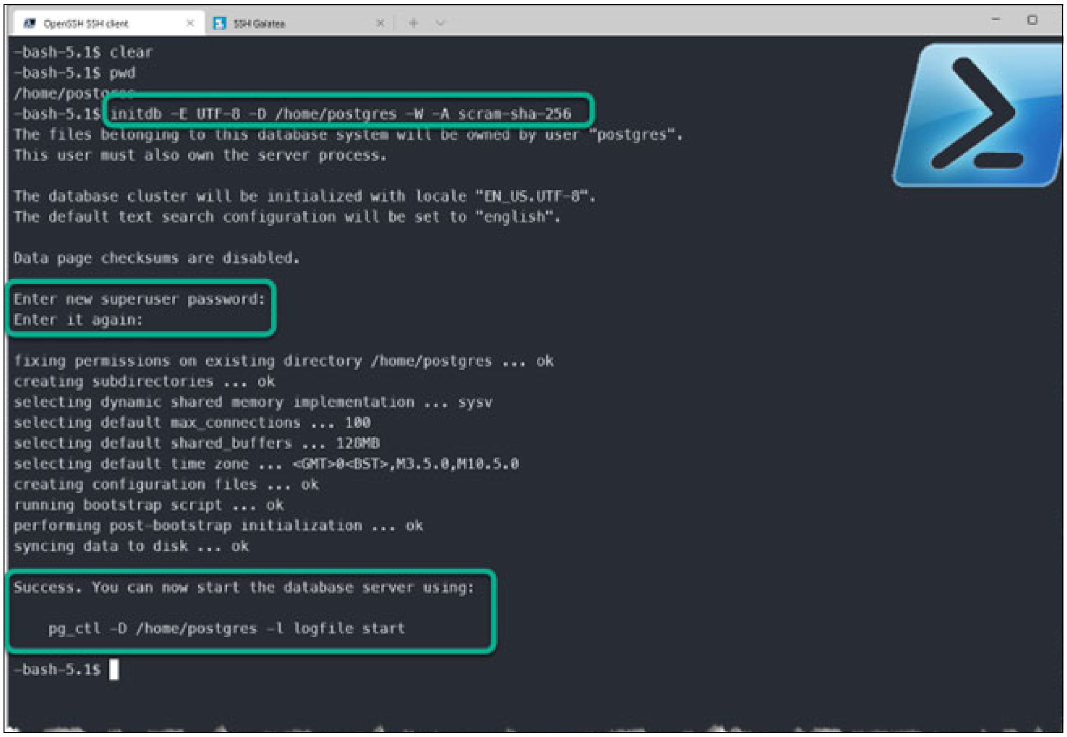
> **Important:** Do not place any files in the *POSTGRES* home directory, it will cause you grief!

3. Then we have to initialize the configuration, using this command.

   ```
   initdb -E UTF-8 -D /home/postgres -W -A scram-sha-256
   ```

   During the configuration, you will have to provide a POSTGRES profile. This is internal for POSTGRES, and nothing to do with the POSTGRES user profile created earlier in this article.

   Figure 9-32 shows the database initialization.



```
-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 ... 128MB
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$ 
```

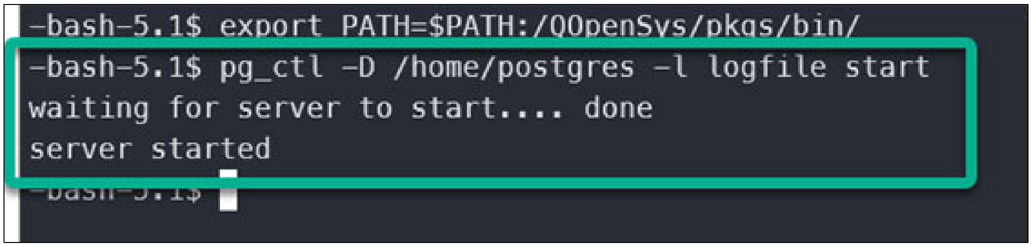
Figure 9-32 Database initialization

Starting the Server

Once the setup has completed, it instructs us to start the server. Use the command below to achieve this.

```
pg_ctl -D /home/postgres -l logfile start
```

The results are shown in Figure 9-33.

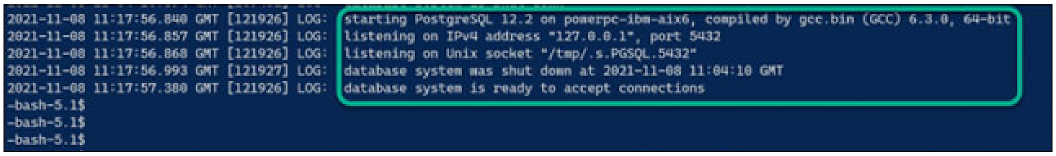


```
-bash-5.1$ export PATH=$PATH:/QOpenSys/pkgs/bin/
-bash-5.1$ pg_ctl -D /home/postgres -l logfile start
waiting for server to start.... done
server started
-bash-5.1$
```

Figure 9-33   Starting the server

The installation is now completed! Our Postgres database server is now up and running.

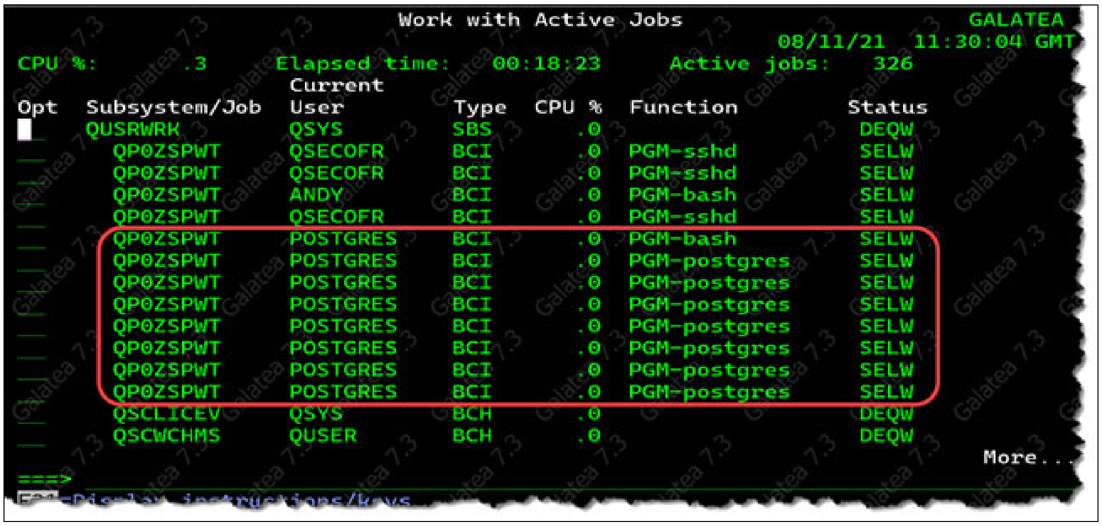
The Postgres server runs on port 5432. This can be checked by viewing the logfile in the /home/postgres directory on our IFS as seen in Figure 9-34.



```
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/.s.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

Alternatively, we can use the *NETSTAT *CNN* command from a 5250 session, as shown in Figure 9-41 on page 322, to check that port *5432* is accepting connections or check the jobs are running in *QUSRWRK* subsystem.



```
                         Work with Active Jobs                      GALATEA
                                                        08/11/21  11:30:04 GMT
CPU %:      .3     Elapsed time:   00:18:23     Active jobs:   326

                             Current
Opt  Subsystem/Job          User        Type  CPU %  Function        Status
 _   QUSRWRK                QSYS        SBS      .0                  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
 __    QSCLICEV             QSYS        BCH      .0                  DEQW
 __    QSCWCHMS             QUSER       BCH      .0                  DEQW
                                                                      More...
===>
F3=Display instructions/keys
```

Figure 9-35   Using NETSTAT command to show PostGres running

Submitting to Batch

If you wish to submit the start server job instead of running it, you can use the submit job command.

```
SBMJOB CMD(QSH CMD('/QOpenSys/pkgs/bin/pg_ctl -D /home/postgres -l logfile
start')) JOB(STR_POSTGR)
```

> **Important:** Ensure you run this command using the *POSTGRES* user profile.

Stopping the Server

For completeness, we can use the command below to stop the server.

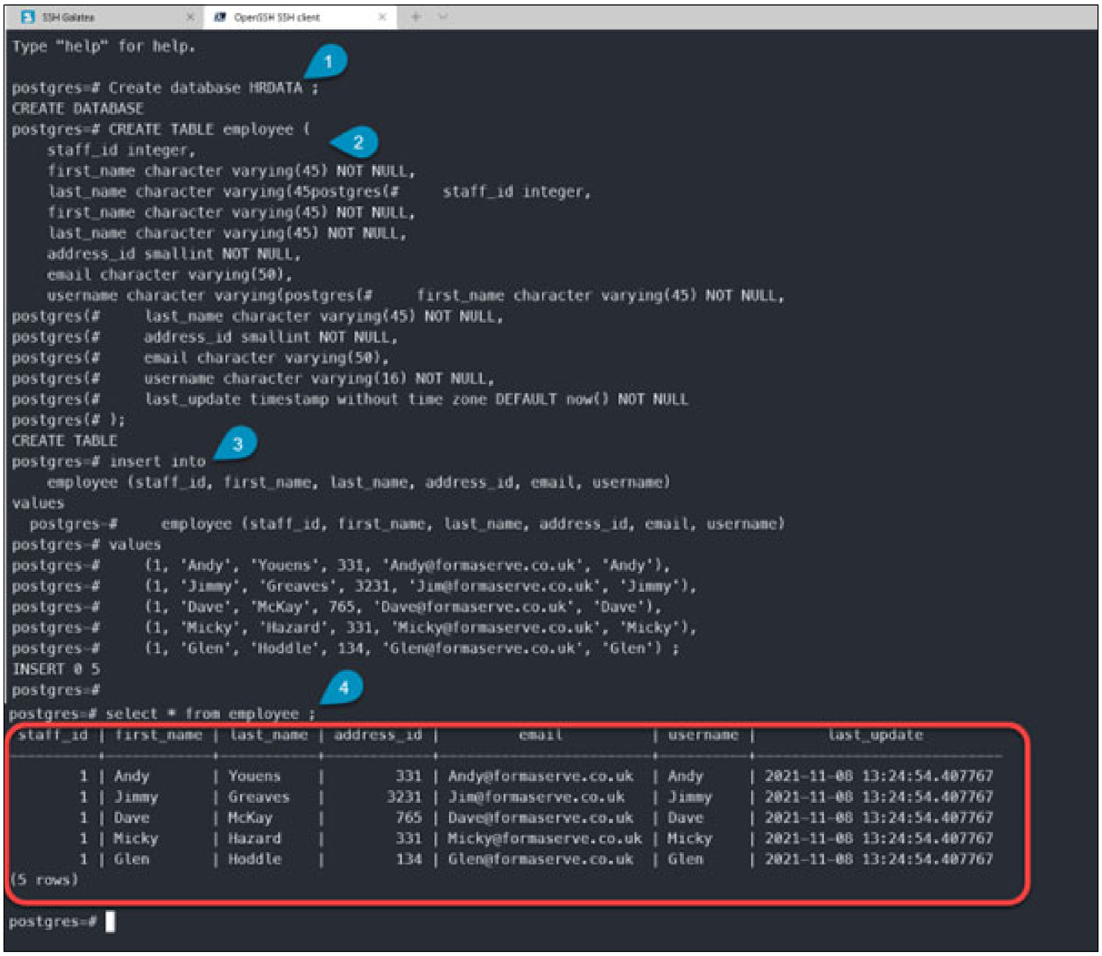
```
pg_ctl -D /home/postgres -l logfile stop
```

Using Postgres

We use the **psql** command to enter SQL statements into Postgres.

In Figure 9-36, we can see the following SQL statements were performed;

1. Create a new database
2. Create a new table
3. Insert some records into the new table
4. Check if the records were inserted



```
Type "help" for help.

postgres=# Create database HRDATA ;
CREATE DATABASE
postgres=# CREATE TABLE employee (
    staff_id integer,
    first_name character varying(45) NOT NULL,
    last_name character varying(45postgres(#     staff_id integer,
    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(postgres(#     first_name character varying(45) NOT NULL,
postgres(#     last_name character varying(45) NOT NULL,
postgres(#     address_id smallint NOT NULL,
postgres(#     email character varying(50),
postgres(#     username character varying(16) NOT NULL,
postgres(#     last_update timestamp without time zone DEFAULT now() NOT NULL
postgres(# );
CREATE TABLE
postgres=# insert into
    employee (staff_id, first_name, last_name, address_id, email, username)
values
  postgres-#     employee (staff_id, first_name, last_name, address_id, email, username)
postgres-# values
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
postgres=#
postgres=# select * from employee ;
 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)

postgres=#
```

Figure 9-36   Validating commands have been run

To quit the **psql** command, use **\q** which will take you back to a Bash session.

9.4.3 SQLite

SQLite is a serverless database, self-contained and open source. It is also referred to as an embedded database which means the database engine runs as a part of your application. There is no central server, so no configuration jobs to do there. The main purpose of SQLite is that it can be embedded in programs. So if you need to store data, on a temporary basis, maybe this database is a better option for your needs.

Installation

Like all the open-source offerings made available by IBM, installation is easy, by using either the Access for Client Solutions (ACS) or using Yum from a SSH shell session.

To show the installation, we will use the yum method here.

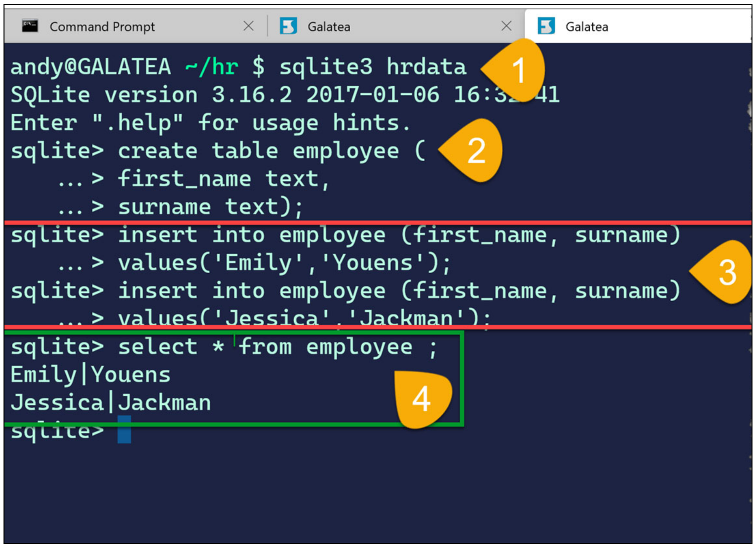
```
yum install sqlite3 sqlite3-devel
```

Using SQLite

To start SQLite, just type SQLite3 and the name of a database you want to work with, into a SSH shell session.

A new database is created if the file does not previously exist.

Figure 9-37 shows creating a new table called employee, in the HRDATA schema and inserting a couple of records.



```
andy@GALATEA ~/hr $ sqlite3 hrdata
SQLite version 3.16.2 2017-01-06 16:32:41
Enter ".help" for usage hints.
sqlite> create table employee (
   ...> first_name text,
   ...> surname text);
sqlite> insert into employee (first_name, surname)
   ...> values('Emily','Youens');
sqlite> insert into employee (first_name, surname)
   ...> values('Jessica','Jackman');
sqlite> select * from employee ;
Emily|Youens
Jessica|Jackman
sqlite>
```

Figure 9-37   Using SQLite

1. Start SQLite and create HRDATA schema
2. Create the employee table
3. Insert records into the employee table
4. List all employee records.

We are now in the database command line interface. Here we can run any standard SQL statement but each SQL statement must end in a semi-colon character.

.DOT Commands

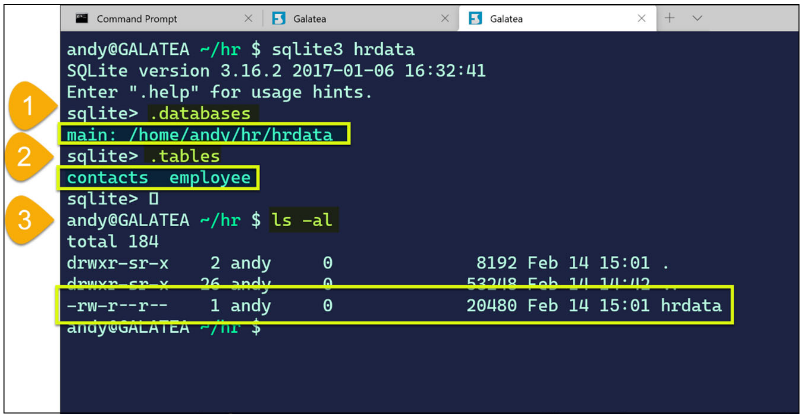
While you are in the SQLITE3 command line, there are numerous commands you can input to show you information about your database. These commands all start with the full stop character.

A summary of these commands are shown in Table . Check out the full list on the official SQLite documentation which can be found here https://www.sqlite.org/cli.html.

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	Has been used once or twice!

Figure 9-38 shows entering a couple of the dot commands, with the output they produce.



```
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> ▯
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 $
```

Figure 9-38 Using commands in SQLite

1. Show me all the databases
2. Show me all the tables in this database
3. Show the actual file that is storing everything in the database. The complete database is held in a single file (hrdata). Makes life very easy for saving and restoring.

9.4.4 Comparison

With the different options for databases on IBM i, you need to consider which database to use. Within your environment, you may have the need for more than one database for different areas of your application suite. You should choose a database based the requirements for that part of your application.

Table 9-3 provides a comparison between MariaDB and SQLite.

Table 9-3 Comparison between MariaDB and SQLite

Comparison	SQLite	MariaDB (MySQL)	Comments
Column types	Only supports: Blob, Integer, Null, Text & Real datatypes	Full set of datatypes as you would expect	As you can tell, MariaDB is a lot more flexible when it comes to data types.

Comparison	SQLite	MariaDB (MySQL)	Comments
Storage and Portability	Stores the database in a single file, making it easily portable, while very small in size. Remember, no configurations are required	With a bigger footprint, MariaDB is easily scalable and can handle a bigger database with less effort	SQLite is suitable for smaller databases, while optimization is easier with MariaDB
Security	SQLite does not have an inbuilt authentication mechanism. The database files can be accessed by anyone.	MariaDB comes with integral security features, which includes authentication.	If you have any security concerns, and need to restrict access, then MariaDB is the only route to take between the two.

SQLite is ideal for

- Developing small standalone apps
- Smaller projects which do not require much scalability
- When you have a requirement to read and write directly from the disk
- Basic development and testing

The availability of SQLite on IBM i is particularly noteworthy. It provides a simpler alternative to more complex database systems, offering a streamlined experience in both development and administration.

SQLite and MariaDB, while both excellent open-source database options, have some key differences in their architecture and features. Ultimately, the best choice for your project depends on your specific needs and priorities. You'll need to weigh the pros and cons of each to determine the right fit. This wealth of open-source database options, including SQLite and MariaDB, gives developers a lot of flexibility.

It's a great advantage to be able to choose the perfect tool for the job, whether that's one of these open-source solutions or even sticking with a robust option like DB2 for i.

9.5 Traditional Programming languages

The IBM i platform, with its long-standing history and reliability, continues to provide a solid foundation through traditional programming languages that have demonstrated their dependability in business environments.

9.5.1 RPG

RPG (Report Program Generator) has long been a cornerstone language in the IBM i ecosystem. Initially developed for generating reports, RPG has evolved significantly, particularly with the introduction of ILE RPG, transforming into a powerful language for building complex business applications. Its ability to efficiently manage data-intensive tasks makes it essential for handling core business processes.

Modernizing RPG

Over the years, RPG has undergone substantial changes, most notably the transition from traditional RPG (fixed-form) to free-form RPG. Fixed-form RPG, with its rigid column-based syntax, required developers to position 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 work with, particularly for modern developers familiar with more flexible programming languages. Despite this, fixed-form RPG proved highly efficient for business data processing and large-scale transaction management, which is why it remains prevalent in many legacy IBM i systems.

The introduction of free-form RPG with RPG IV marked a major step in modernizing the language. Free-form RPG eliminates the need for column-based formatting, enabling developers to write code more freely, similar to modern programming languages like C, Java, or Python. There are no column restrictions in free-form RPG, and statements such as IF, DO, and FOR are written in a more intuitive, readable manner. This shift makes the code easier to maintain and understand. Additionally, free-form RPG simplifies the integration with embedded SQL, allowing for more seamless interaction with DB2 for i databases. It also supports modern programming practices, including modular programming and improved error handling, which enhances the scalability and maintainability of complex applications.

The transition to free-form RPG reflects IBM’s efforts to make the IBM i platform more accessible to a new generation of developers, all while preserving its legacy strengths. Free-form RPG has significantly boosted developer productivity by enabling cleaner, more readable code and providing a better experience with modern development tools. Furthermore, it supports greater integration with contemporary software practices such as version control and IDEs, while also offering a more natural approach to implementing features like data structures and error handling. As businesses continue to modernize their systems, the adoption of free-form RPG ensures that IBM i remains adaptable and relevant in the rapidly evolving landscape of enterprise software development.Another key language in the IBM i ecosystem is COBOL (Common Business-Oriented Language). Renowned for its clarity and effectiveness in financial and administrative functions, COBOL remains central to many legacy systems that handle critical business data. While often linked to older applications, its stability and proven reliability keep it highly relevant. The seamless operation of COBOL on the IBM i highlights the platform's dedication to maintaining backward compatibility, preserving valuable business logic.

Conversion from fixed-form RPG to free-form RPG

IBM provides conversion tools that help automate the migration of fixed-form RPG code to free-form RPG. These tools analyze the existing code and help with the initial conversion process. However, developers still need to review and test the converted code to ensure that it functions correctly.

Converting from fixed-form RPG to free-form RPG is a worthwhile step toward modernizing your IBM i applications. By removing column restrictions, improving code readability, and supporting contemporary development practices, free-form RPG makes it easier for developers to maintain and extend IBM i systems. The conversion process involves understanding the differences in syntax and making the code more flexible, intuitive, and integrated with modern development tools. While the conversion can require effort, the long-term benefits in terms of productivity and system maintainability are significant.

9.5.2 COBOL

In addition to these application-focused languages, CL (Control Language) plays a crucial role in system administration and operational control on IBM i. CL is used for task automation, job management, and interacting with the operating system, offering essential tools for system administrators to maintain and optimize the platform. Its significance lies in its ability to streamline operations and ensure the efficient functioning of IBM i environments.

Together, these traditional programming languages — RPG, COBOL, and CL — have shaped the IBM i platform and remain vital to many businesses today. Their enduring presence reflects their continued reliability and efficiency in supporting business-critical applications. While the IBM i also supports modern languages, these traditional tools continue to provide a stable and proven 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 be used to compliment and enhance the traditional programming languages that have been used for many years.

Figure 9-39 shows the time line for open-source programming languages and frameworks were first added to IBM i.

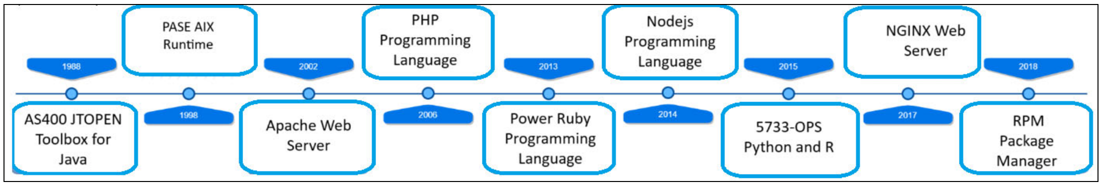


Figure 9-39 Open-source time line on IBM i

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. Node.js runs the V8 JavaScript engine outside of the browser. Node.js was first introduced to the IBM i in 2014, as part of the 5733-OPS licensed program before moving to the RPM installation process.

Python

Python is a high-level, interpreted programming language known for its readability and simplicity, and has grown to become one of the most popular programming languages. It is a flexible language that supports multiple programming models, including procedural, object-oriented and functional programming.

PHP

A very widely used open-source scripting language specifically designed for web development. It has become one of the foundational languages of the web. It is embedded within HTML and is particularly suited for server-side development, enabling the creation of dynamic web pages and applications. PHP is known for its versatility and ease of integration with various databases such as MySQL, PostgreSQL and SQLite.

PHP has had a long, and eventful, lifespan on the IBM i:

2002	Unsupported versions of PHP found their way onto the server

2006	Zend release a version of their Zend Server product, that was installed as a licensed program. Apart from the basic license, this was a chargeable feature.
2018	IBM and Zend released community PHP, an unsupported open-source offering installed via the RPM package manager

Ruby

Ruby is a dynamic, open-source programming language with a focus on simplicity and productivity. Ruby is best known for its powerful web application framework, Ruby on Rails, which has significantly contributed to the language's popularity.

It has been available on the IBM i since 2013 and was originally available as part of the 5773-OPS licensed program structure.

R is a software environment specifically designed for statistical computing and graphics. The R programming language is remarkable for performing data analysis for business insights. It has been available on the IBM i since 2015 and was originally available as part of the 5773-OPS licensed program structure.

It is not widely used on the IBM I, nor widely supported by IBM.

Java

Java is a high-level, object-oriented programming language developed by Sun Microsystems (now part of Oracle Corporation), which is known for its 'write once, run anywhere' (WORA) capability.

It has been available since the introduction of the AS400 in 1988, via the JTOpen Toolbox for Java. This was a traditional licensed program.

9.6.2 Open-source Programming Support

IBM provides support for 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 suppling open-source from the 5733-OPS licensed program,

Additionally, IBM provides some support for integration packages. These include the following packages:

- iToolkit
- idb-connector
- idb-pconnector
- loopback-connector-IBM i
- ODBC

Also supported are these Python packages:

- toolkit
- ibm_db

Full details on IBM support for open-source, can be found at
https://www.ibm.com/support/pages/open-source-support-ibm-i

9.7 Integrating traditional and open-source languages

In this section, we will explore how to integrate traditional programming languages used on the IBM i platform with open-source languages. We'll demonstrate how combining these approaches allows us to leverage the best of both worlds.

While IBM has introduced a wide array of open-source languages, their adoption by organizations is often dependent on the ability to integrate both traditional and open-source languages effectively. By doing so, enterprises can fully maximize the benefits of modern technologies.

In today's rapidly evolving tech landscape, combining traditional programming languages with open-source options on the IBM i platform is increasingly essential. This fusion enables businesses to preserve their past investments while embracing the flexibility and capabilities offered by open-source solutions. For organizations aiming for enhanced functionality, adaptability, and modernization, integrating both worlds is a critical strategy.

We will provide examples of the following:

Traditional Languages connecting to Open-Source Languages

For traditional languages using open-source extensions we provide examples of:

- Using Control Language (CL) to:
  - Call a Node.js script
  - Call a PHP script
  - Call a Python script
- Using RPG (Free Format) to:
  - Call a Node.js script
  - Call a PHP script
  - Call a Python script

Open-Source languages connecting to Traditional Languages

For writing new open-source routines that connect to traditional languages we provide examples of:

- Using Node.js to:
  - Call service programs
  - Execute IBM i commands
- Using PHP to:
  - Call service programs
  - Execute IBM i commands
- Using Python to:
  - Call CL programs
  - Execute IBM i commands

The ability to combine new technologies with existing application code enables organizations to extend and modernize their legacy systems. By integrating newer programming languages, businesses can breathe new life into their applications while maintaining the core functionality. This approach also opens up opportunities to tap into the expertise of developers who may not be familiar with traditional IBM i programming languages, allowing for greater flexibility and innovation in development.

Important considerations

For our examples you should understand:

- In this section, all examples make use of environment variables to store sensitive information, ensuring that the authors' security values are not exposed. This approach allows for practical demonstrations while maintaining best practices in data security.
- The section exclusively focuses on the latest version of RPG IV Free Format, as this is the preferred version for modernization tasks and projects.

Language versions and terminology

All code snippets contained in this section have been verified with the languages shown in Table 9-4. Throughout this section we will refer to each language with the short name shown in the table.

Table 9-4 Programming languages covered

Name	Short Name	Version
Command Language	CL	7.5
RPG IV Free format	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 the connectivity from traditional languages to open-source routines.

Command Language (CL)

Our first examples utilize IBM i command language to call open-source routines.

CL to Node.js

In this first example in the Control Language section, we will show a simple CL program executing a Node.js script. This uses the **QSH** command to start a shell session. QSH is an IBM i command and is built into operating system. Example 9-14 shows the CL script.

Example 9-14 CL script to call 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/pkgs/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 integrated file system (the file name is included in the variable VALUE in the CL script. The Node.js script contents are shown in Example 9-15.

Example 9-15 Node.js script called by CL

```
console.log("Hello from Node.js!");
```

A very simple example to start with, but we can see all the main principles here. The Node script will output to the joblog *"Hello from Node.js!"*, which proves we can execute scripts that reside within our Integrated File System (IFS).

CL to PHP

In our next example we build on the concept above and introduce parameters that are passed from the CL program to the PHP script. Please note that in the open-source world, parameters are often called arguments or options. The CL program is shown in Example 9-16.

Example 9-16 CL to call 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
```

And to follow, the PHP script is shown in Example 9-17.

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";
```

To call the CL program see Example 9-18.

Example 9-18 Calling the command line 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 examples above to get you started. The CL script is shown in Example 9-19.

Example 9-19 CL script to call 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
```

The python example is shown in Example 9-20.

Example 9-20 Python routine 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}")
```

The output is shown in Example 9-21.

Example 9-21 Output from Python routine

```
Argument 1: hello
Argument 2: world
```

RPG

These examples are created using RPG to call different open-source routines.

RPG calling a node script

There are many occasions where it is necessary to get RPG to call a Node.js script and receive the values returned.

Example 9-22 shows how a simple RPG program can call a node 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 execute
   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/pkgs/bin/node /home/user/node1.js'')';
// Execute the command using QCMDEXC
QCMDEXC(Cmd: %len(%trim(Cmd)));
*inlr = *on; // End program
```

Example 9-23 shows the node script that is being called.

Example 9-23 The node.js script called

```
// node1.js
console.log("Hello from Node.js!");
```

RPG calling a PHP script

Next, we move onto RPG calling a PHP script. The CL script is shown in Example 9-24.

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/pkgs/bin/php /home/user/redbook/php1.php ' +
  %trim(Param1) + ' ' + %trim(Param2);
cmdstring2 = 'QSH cmd(' + quote + %trim(cmdstring1) + quote + ')' ;
CmdLength = %len(%trimr(CmdString2));
// Executing the command
QCMDEXC(CmdString2: CmdLength);
*inlr = *on; // End program 'omers
```

Example 9-25 shows the PHP script being 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 the Python script using the QSH (QShell) command. It passes the parameters param1 and param2 to the Python script. The Python script then retrieves the parameters from the command line arguments and processes them accordingly. The RPG code is shown in Example 9-26.

Example 9-26 RPG code to call 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));
// Executing the command
QCMDEXC(CmdString2: CmdLength);
*inlr = *on; // End program 'omers
```

The Python script is shown in Example 9-27.

Example 9-27 Python script 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 Toolkit is an interface that makes it easy to connect modern web apps with the IBM i. This toolkit helps traditional systems to create new web services and applications that can easily work with IBM i data and processes. With the iToolkit, you can streamline operations and explore new possibilities for innovation on the IBM i platform. The toolkit is installed using the NPM (Node Package Manager) install command: **`npm install itoolkit`**.

The first thing we must do when using the toolkit is to create a connection to our IBM i.

The supported transport methods for creating a connection are;

ODBC The ODBC transport is used to call DB2 for i stored procedures. This is a generic connection and can also be used to other (non-IBM i) servers.

SSH The SSH transport is used make a secure shell connection

idb This service is used as an alternative to the ODBC system, but can only be used for connections to an IBM i server

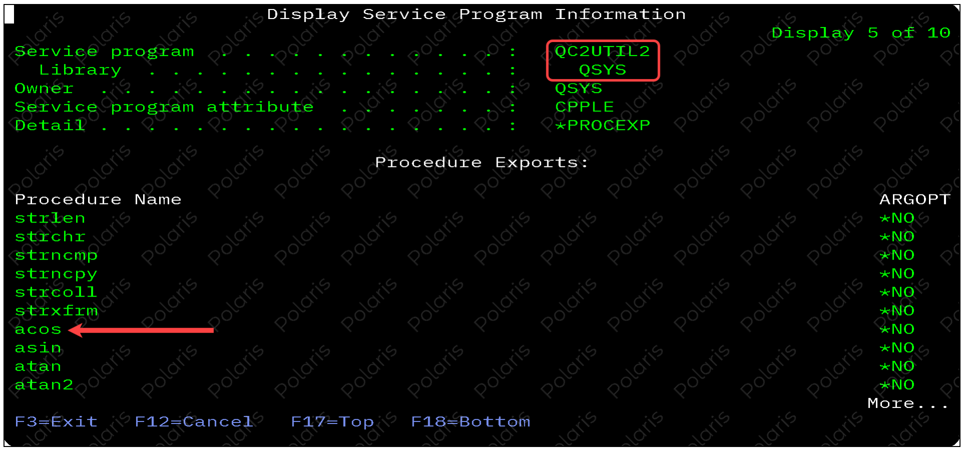
REST This connection makes an HTTP request to a REST API endpoint

The examples shown in the rest of this section will be made using the SSH transport method. If you wish, or need, to connect using other transport methods, please refer to the IBM Open-Source documentation.

Node Calling a Service Program

In this example we show how easy it is to integrate Node.js and the more traditional world of service programs and programs, whether they be RPG, Cobol or CL programs.

In the example below, a Node.js program calls an IBM service program called **QC2UTIL2**. This service program is part of the IBM i operating system and is available to use on all software versions of the IBM i. As can be seen in Figure 9-40, the **QC2UTIL2** has many procedures that are exported and available to other programs. We will be using the ***acos*** procedure as an example to show how we can pass parameters to it and receive a returned value.



```
                      Display Service Program Information
                                                               Display 5 of 10
 Service program . . . . . . . . . . . . . :   QC2UTIL2
   Library . . . . . . . . . . . . . . . . :     QSYS
 Owner . . . . . . . . . . . . . . . . . . :   QSYS
 Service program attribute . . . . . . . . :   CPPLE
 Detail . . . . . . . . . . . . . . . . . . :   *PROCEXP

                              Procedure Exports:

 Procedure Name                                                          ARGOPT
 strlen                                                                  *NO
 strchr                                                                  *NO
 strncmp                                                                 *NO
 strncpy                                                                 *NO
 strcoll                                                                 *NO
 strxfrm                                                                 *NO
 acos                                                                    *NO
 asin                                                                    *NO
 atan                                                                    *NO
 atan2                                                                   *NO
                                                                       More...
 F3=Exit   F12=Cancel   F17=Top   F18=Bottom
```

Figure 9-40 Capabilities of QC2UTIL2

This example uses the ***fast-xml-parser*** module to convert iToolkit's raw XML output to JSON, which the Node.js script can read natively. This is automatically installed as a dependency of the latest iToolkit package, and NPM performs the install automatically.The Node.js script is shown in Example 9-28.

Example 9-28 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 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 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
});
```

The results from running this example program are shown in Example 9-29.

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 the debug is switched off, the only output we would see is the return value of 1.

Node.js Executing IBM i Commands

The next example demonstrates how to run an IBM i command using the Node.js iToolkit We will run the command to retrieve the current jobs library list for both the user and the system parts. The command used is **RTVJOBA** (Retrieve Job Attributes). The Node.js program is shown in Example 9-30.

Example 9-30 Node.js program to execute 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 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 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
});
```

The results are shown in Example 9-31.

Example 9-31 Results of running 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 we use Python to connect to traditional IBM i functions.

Calling Programs

In this example we will demonstrate how to use python to call an RPG program with two parameters, one for input, the other for the output that is returned to the python script.

The python code is shown in Example 9-32. It includes error handling, for ‘just in case’.

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 (e.g., 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 the program can access required library resources
    itool.add(iCmd('addlible', 'addlible 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
    )
    # Executing 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 executed 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))
```

The RPG source is shown in Example 9-33.

Example 9-33   RPG source

```
**free
// Take one parameter from a python program & return another
// Nothing fancy, but 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
```

The output from running the python program is shown in Example 9-34.

Example 9-34   Output from 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 Execute IBM i Commands

Here we can see a typical python script that executes an IBM i command. It uses a database connection, then executes the QCMDEXC stored procedure to execute the IBM i command. All very simple and easy to use. The python code is shown in Example 9-35.

Example 9-35   Python code to execute an IBM i command

```
# This is a simple example of how to call a stored procedure in DB2 for i 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}")
```

Next, we have another example of a different method of executing an IBM i command.This example uses the iToolkit to create a database connection that we use to execute commands. The code is shown in Example 9-36.

Example 9-36 Example python code to execute 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 execute 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

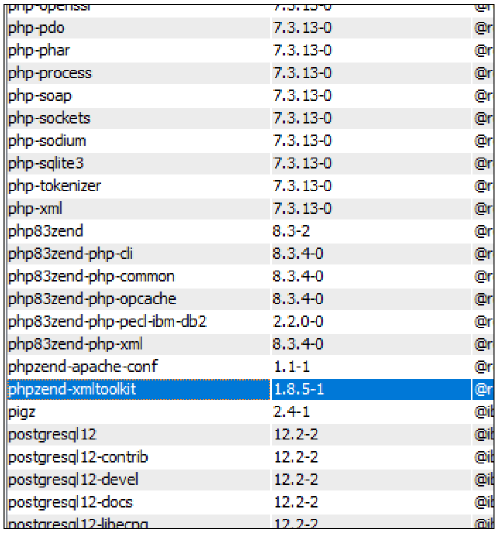
This section uses PHP to connect to an IBM i system and run commands and programs.

PHP Calling a RPG Program

This section provides an example of how a PHP script can call an IBM i RPG program. The communication is performed using the Toolkit for IBM i, a PHP library that provides an easy interface to call programs and use IBM i resources. The Toolkit needs to be installed on your IBM i before attempting to use any of the examples included within this section.

The toolkit is installed as part of the Open-Source Package Management utility, which itself is part of the Access for Client Solutions (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-xmltoolkit	1.8.5-1	@r
pigz	2.4-1	@ib
postgresql12	12.2-2	@ib
postgresql12-contrib	12.2-2	@ib
postgresql12-devel	12.2-2	@ib
postgresql12-docs	12.2-2	@ib
postgresql12-libecpg	12.2-2	@ib

Figure 9-41 List of programs in IBM i Open Source Package Management

The PHP source is shown in Example 9-37.

Example 9-37 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 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', 'REDBOOK', $Params);
    // Display the result
    echo 'Result: ' . $outputParams['RESULT'] . "\n";
} catch (Exception $e) {
    // Handle exceptions
    echo 'Error: ' . $e->getMessage() . "\n";
}
```

The RPG that is called by the PHP is the same program that is used in the python script to call a RPG program. This can be found in Example 9-33 on page 320.

PHP Executing a IBM i Command

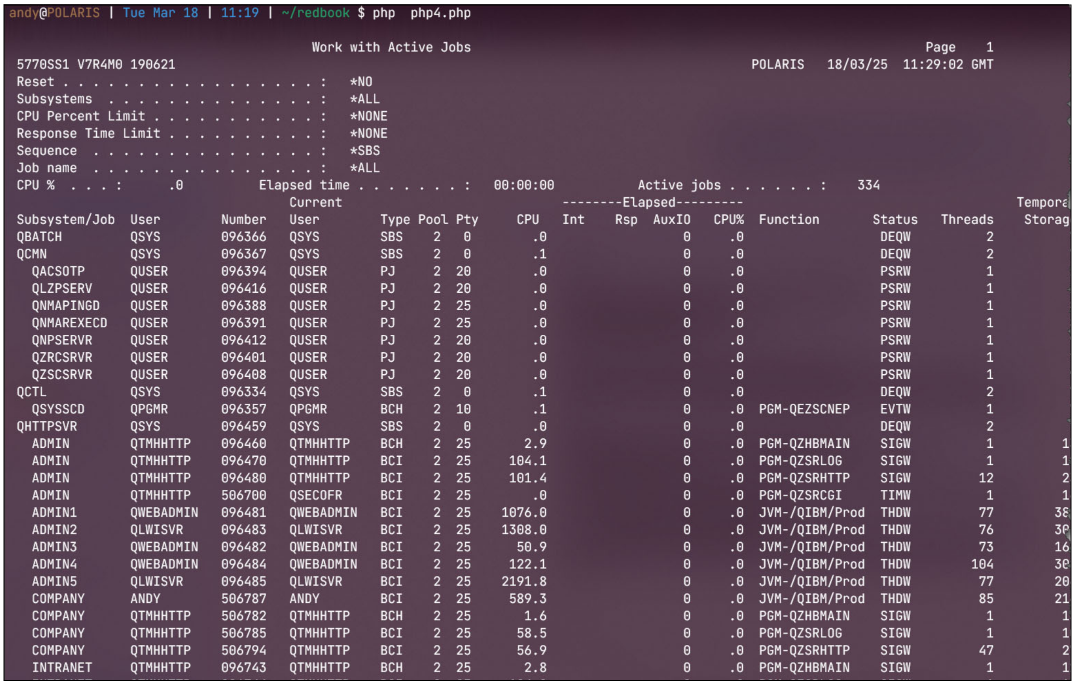
Below is an example of how a PHP script can execute an IBM i command. In this example we will execute the Work Active Jobs command using the database transport mechanism from the PHP iToolkit.

The PHP source is shown in Example 9-38.

Example 9-38   PHP source for calling IBM i command

```
<?php
// This script connects to an IBM i system 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 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 is executed 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)
```

I'm sure you can guess the output from this example, but for completeness it is shown in Figure 9-42.



```
andy@POLARIS | Tue Mar 18 | 11:19 | ~/redbook $ php  php4.php

                                        Work with Active Jobs                                                      Page    1
 5770SS1 V7R4M0 190621                                                                       POLARIS   18/03/25  11:29:02 GMT
 Reset . . . . . . . . . . . . . . . . . . :   *NO
 Subsystems  . . . . . . . . . . . . . . . :   *ALL
 CPU Percent Limit . . . . . . . . . . . . :   *NONE
 Response Time Limit . . . . . . . . . . . :   *NONE
 Sequence  . . . . . . . . . . . . . . . . :   *SBS
 Job name  . . . . . . . . . . . . . . . . :   *ALL
 CPU %  . . . :      .0          Elapsed time . . . . . . . :   00:00:00          Active jobs . . . . . . :    334
                                     Current                                --------Elapsed---------                                  Tempora
 Subsystem/Job  User         Number   User         Type Pool Pty     CPU   Int     Rsp  AuxIO   CPU%  Function        Status   Threads   Storag
 QBATCH         QSYS         096366   QSYS         SBS     2   0      .0                    0     .0                  DEQW           2
 QCMN           QSYS         096367   QSYS         SBS     2   0      .1                    0     .0                  DEQW           2
   QACSOTP      QUSER        096394   QUSER        PJ      2  20      .0                    0     .0                  PSRW           1
   QLZPSERV     QUSER        096416   QUSER        PJ      2  20      .0                    0     .0                  PSRW           1
   QNMAPINGD    QUSER        096388   QUSER        PJ      2  25      .0                    0     .0                  PSRW           1
   QNMAREXECD   QUSER        096391   QUSER        PJ      2  25      .0                    0     .0                  PSRW           1
   QNPSERVR     QUSER        096412   QUSER        PJ      2  20      .0                    0     .0                  PSRW           1
   QZRCSRVR     QUSER        096401   QUSER        PJ      2  20      .0                    0     .0                  PSRW           1
   QZSCSRVR     QUSER        096408   QUSER        PJ      2  20      .0                    0     .0                  PSRW           1
 QCTL           QSYS         096334   QSYS         SBS     2   0      .1                    0     .0                  DEQW           2
   QSYSSCD      QPGMR        096357   QPGMR        BCH     2  10      .1                    0     .0  PGM-QEZSCNEP    EVTW           1
 QHTTPSVR       QSYS         096459   QSYS         SBS     2   0      .0                    0     .0                  DEQW           2
   ADMIN        QTMHHTTP     096460   QTMHHTTP     BCH     2  25     2.9                    0     .0  PGM-QZHBMAIN    SIGW           1         1
   ADMIN        QTMHHTTP     096470   QTMHHTTP     BCI     2  25   104.1                    0     .0  PGM-QZSRLOG     SIGW           1         1
   ADMIN        QTMHHTTP     096480   QTMHHTTP     BCI     2  25   101.4                    0     .0  PGM-QZSRHTTP    SIGW          12         2
   ADMIN        QTMHHTTP     506700   QSECOFR      BCI     2  25      .0                    0     .0  PGM-QZSRCGI     TIMW           1         1
   ADMIN1       QWEBADMIN    096481   QWEBADMIN    BCI     2  25  1076.0                    0     .0  JVM-/QIBM/Prod  THDW          77        38
   ADMIN2       QLWISVR      096483   QLWISVR      BCI     2  25  1308.0                    0     .0  JVM-/QIBM/Prod  THDW          76        30
   ADMIN3       QWEBADMIN    096482   QWEBADMIN    BCI     2  25    50.9                    0     .0  JVM-/QIBM/Prod  THDW          73        16
   ADMIN4       QWEBADMIN    096484   QWEBADMIN    BCI     2  25   122.1                    0     .0  JVM-/QIBM/Prod  THDW         104        30
   ADMIN5       QLWISVR      096485   QLWISVR      BCI     2  25  2191.8                    0     .0  JVM-/QIBM/Prod  THDW          77        20
   COMPANY      ANDY         506787   ANDY         BCI     2  25   589.3                    0     .0  JVM-/QIBM/Prod  THDW          85        21
   COMPANY      QTMHHTTP     506782   QTMHHTTP     BCH     2  25     1.6                    0     .0  PGM-QZHBMAIN    SIGW           1         1
   COMPANY      QTMHHTTP     506785   QTMHHTTP     BCI     2  25    58.5                    0     .0  PGM-QZSRLOG     SIGW           1         1
   COMPANY      QTMHHTTP     506794   QTMHHTTP     BCI     2  25    56.9                    0     .0  PGM-QZSRHTTP    SIGW          47         2
   INTRANET     QTMHHTTP     096743   QTMHHTTP     BCH     2  25     2.8                    0     .0  PGM-QZHBMAIN    SIGW           1         1
```

Figure 9-42   Output from wrkactjob

10

Linux and OpenShift

Linux on IBM Power is a robust and adaptable computing platform that seamlessly merges the open-source flexibility of Linux with the power, reliability, and scalability of IBM's Power architecture. Over the years, this combination has evolved significantly, delivering substantial benefits for a wide range of workloads.

IBM's involvement with Linux began in the late 1990s as the operating system's influence in the tech world continued to grow. In 2000, IBM publicly committed to supporting Linux, and by 2001, it made a major investment of $1 billion to advance Linux on its systems. This effort included optimizing the Linux kernel and software to work efficiently on IBM's hardware. Various Linux distributions, such as Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), and Ubuntu, are now supported on IBM Power systems, with custom enhancements tailored for the Power architecture. IBM also works closely with these vendors to ensure consistent and well-supported operating system environments.

For organizations seeking high-performance, reliable, scalable, and secure platforms for their critical workloads, Linux on IBM Power offers a compelling solution. Its rich history of innovation and open collaboration continues to make it a powerful choice in the ever-evolving computing landscape.

Additionally, Linux serves as an excellent foundation for modern workloads and powers Red Hat OpenShift, which provides a Kubernetes-based containerization solution for deploying hybrid cloud applications on IBM Power. Leveraging both Linux and Red Hat OpenShift allows businesses to run hybrid cloud applications on the hardware platform best suited to their specific needs.

The following topics are covered in this chapter:

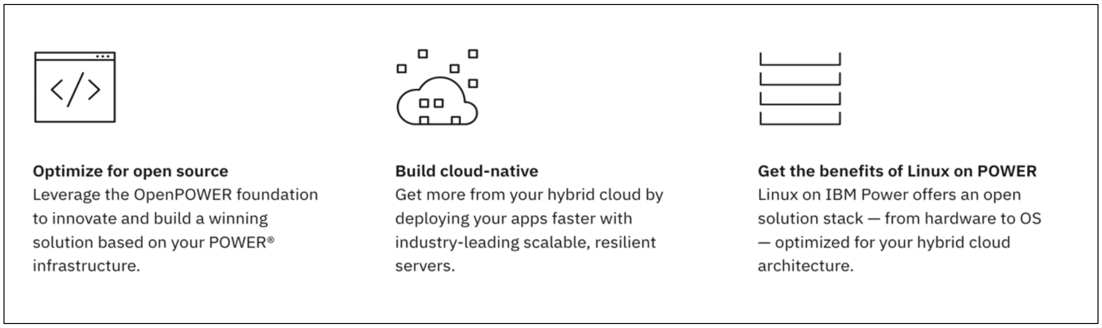
© Copyright IBM Corp. 2025.

10.1 Linux

Linux is an open-source, cross-platform OS that runs on numerous platforms from embedded systems to mainframe computers. It provides an UNIX like implementation across many computer architectures.

IBM Power servers provide unique capabilities to run enterprise Linux distributions with a fully open stack that benefits from the OpenPower ecosystem and efficient cloud-native performance through PowerVM virtualization technology. These servers amplify 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 your open-source hybrid cloud infrastructure, empowering you to modernize applications more efficiently.

Linux on IBM Power provides you with the blend of agility, security and resiliency you need to stay ahead. Figure 10-1 shows some of the benefits of running Linux on IBM Power.



Optimize for open source
Leverage the OpenPOWER foundation to innovate and build a winning solution based on your POWER® infrastructure.

Build cloud-native
Get more from your hybrid cloud by deploying your apps faster with industry-leading scalable, resilient servers.

Get the benefits of Linux on POWER
Linux on IBM Power offers an open solution stack — from hardware to OS — optimized for your hybrid cloud architecture.

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 allows the application to meet your current and future needs. On IBM Power, you can continue running your existing applications while you start surrounding them with new cloud-native applications.

Application modernization opens the door to several business and technical benefits for your organization. Let’s take a closer look at some of them.

- Accelerate digital transformation

  More than ever, organizations need to find new ways to provide innovative, engaging experiences that satisfy existing customers, attract new ones and gain a competitive edge. A Forrester Consulting study – commissioned by IBM – on the business value of modernizing applications with IBM and Red Hat solutions found that modernization efforts help accelerate release frequency by up to 10 times, improving customer engagement, time to market and operations.1

- Gain a superior developer experience

  Your organization’s most valuable assets are its people. When it comes to uncovering hidden competitive advantages through IT, you want to ensure your application developers always have the right set of technologies—and the most up-to-date applications—at their fingertips to unleash their creativity and build truly amazing customer experiences.

- Deploy enterprise applications across your hybrid multicloud

  As enterprises further embrace a hybrid cloud strategy, it’s critically important that applications have the flexibility to be deployed anywhere across this landscape to reap the full benefits. This flexibility allows you to use the continuous innovation that’s happening across public cloud providers along with the security, data privacy and reliability of your own data center. This level of choice and flexibility is paramount for successful competitive differentiation in today's market.

10.1.1 Linux Distributions supported on IBM Power

IBM Power supports a variety of Linux distributions, each offering unique features and capabilities. This section provides an overview of the supported distributions.

Red Hat-Based Distributions

Red Hat Enterprise Linux (RHEL) and its derivatives, CentOS, Fedora, Alma and Rocky Linux, are widely used distributions known for their stability, security, and enterprise-grade support for different use cases from classic workloads such as application servers and big databases to containers, machine learning and others.

Red Hat Enterprise Linux (RHEL) operating system on IBM Power systems allows users to leverage the capabilities of a robust enterprise Linux distribution on high-performance IBM Power hardware, ideal for demanding workloads like big data processing and large-scale virtualization; essentially combining the flexibility of open-source Linux with the power and reliability of IBM Power architecture.

Red Hat Enterprise Linux CoreOS (RHCOS) which is based on Red Hat Enterprise Linux (RHEL) is the next generation of single-purpose container operating system technology. It provides the same quality standards of Red Hat Enterprise Linux (RHEL) with automated, remote upgrade features. RHCOS is supported only as a component of OpenShift Container Platform for all OpenShift Container Platform machines.

Fedora CoreOS is an automatically updating, minimal, monolithic, container-focused operating system. It is designed for clusters but can also operate as standalone OS. Fedora CoreOS is optimized for Kubernetes and acts as a container host to run containerized workloads securely and at scale. For more information about Red Hat Enterprise Linux see this Red Hat website.

SUSE-Based Distributions

SUSE Linux Enterprise Server (SLES) traditionally used for SAP HANA on Power environments can also be used for classic workloads. In addition, OpenSUSE Leap is a community-driven, open-source Linux distribution, developed by the OpenSUSE Project. It shares its core with SUSE Linux Enterprise (SLE), providing a highly stable and well-tested base. receives the same security fixes as soon as they are released to SLE customers.

SUSE Linux Enterprise Server for IBM POWER is an enterprise-grade Linux distribution optimized for IBM POWER-based systems. It is designed to deliver increased reliability and provide a high-performance platform to meet increasing business demands and accelerate innovation while improving deployment times. For more information about SUSE Linux Enterprise Server for IBM Power see this website: https://www.suse.com/products/power/.

Debian-based distributions

Debian is a popular and widely-used operating system, primarily known for its stability, reliability, security and extensive software repositories. It is a Linux distribution consisting entirely of free software. Debian is the foundation for many other distributions, most notably Ubuntu also supported on Power.

Ubuntu is optimized for workloads in the mobile, social, cloud, Big Data, analytics and machine learning spaces. With its unique deployment tools (including Juju and MAAS), Ubuntu makes the management of those workloads trivial. Starting with Ubuntu 22.04 LTS, POWER9 and POWER10 processors are supported. For more information about Ubuntu Server see this website: https://ubuntu.com/server

> For information on how to install Linux on IBM Power10 systems refer to https://www.ibm.com/docs/en/linux-on-systems?topic=linux-installing-power10-systems

Linux support on Power10 servers

As discussed earlier, multiple Linux distributions are supported on IBM Power servers. Table 10-1 captures 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 LPARs
► 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)	► Red Hat Enterprise Linux 9.0, any subsequent RHEL 9.x releases ► Red Hat Enterprise Linux 8.4, any subsequent RHEL 8.x releases ► SUSE Linux Enterprise Server 15 SP3, any subsequent SLES 15 updates ► Red Hat OpenShift Container Platform 4.9, or later ► Ubuntu 22.04, or later [a]
► 9080-HEX (IBM Power E1080)	► Red Hat Enterprise Linux 9.0, any subsequent RHEL 9.x releases ► Red Hat Enterprise Linux 8.4, any subsequent RHEL 8.x releases ► Red Hat Enterprise Linux 8.2 (POWER9 Compatibility mode only)[b] ► SUSE Linux Enterprise Server 15 SP3, any subsequent SLES 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)	► Red Hat Enterprise Linux 9.2, for PowerLE, or later ► Red Hat OpenShift Container Platform 4.15, or later ► SUSE Linux Enterprise Server 15 SP6, any subsequent SLES 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.
- Red Hat Enterprise Linux 8.6, or later: up to 64 TB of memory and 240 processor cores.
- Red Hat Enterprise Linux 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 recommended Linux distribution for a particular server is always the latest level distribution that is optimized for the server. The listed distributions are the operating system versions that are supported for the specific hardware. For information about product lifecycle for Linux distributions, see the support site for each distribution.

SUSE Linux Enterprise Server	`https://www.suse.com/lifecycle/`
Red Hat Enterprise Linux	`https://access.redhat.com/support/policy/updates/errata`
Ubuntu	`https://ubuntu.com/about/release-cycle`

> **Note:** CoreOS is supported as a part of OpenShift Container Platform (OCP). For more information about OCP, see Getting started with Red Hat OpenShift on IBM Cloud and Red Hat OpenShift Container Platform.

To learn more about Linux on Power and various supported combinations, please refer to official IBM documentation.

Now that we have talked about supported Linux releases on IBM Power, lets understand some of the important features and functionality supported with Linux on IBM Power.

10.1.2 Linux on Power Features and Functionality

Linux on Power includes specific support for features in the IBM Power platform. We describe some of these in this section.

Hybrid Network Virtualization (HNV)

Hybrid Network Virtualization enhances the availability and performance of your Linux partitions on IBM Power servers by leveraging two key features: Single Root I/O Virtualization (SR-IOV) and Live Partition Mobility (LPM).

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

  SR-IOV allows a single I/O adapter to be shared simultaneously across multiple logical partitions, providing hardware-level speeds without adding CPU overhead. This is because the I/O adapter itself handles virtualization at the hardware level. Before SR-IOV, each network adapter was dedicated to a single partition and required virtualization via the Virtual I/O Server (VIOS) to be shared across multiple partitions. As network speeds increased, managing virtualized network traffic on the VIOS demanded more compute resources, which introduced additional overhead and often increased network latency. With SR-IOV, network adapters are simplified, providing improved performance and reduced VIOS overhead. However, this setup limited the ability of partitions to benefit from another important IBM PowerVM feature: Live Partition Mobility.

- Live Partition Mobility (LPM)

  IBM PowerVM’s Live Partition Mobility (LPM) feature enables the seamless relocation of a running virtual partition from one physical IBM Power server to another. This capability is especially useful for preventing service disruptions during planned events, such as maintenance or upgrades, allowing for the complete evacuation of a server without interrupting workloads. In addition to planned downtime, LPM helps with dynamic workload balancing, optimizing resource utilization 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 were unable to leverage LPM's benefits. To solve this, IBM developed Host Network Virtualization (HNV), which virtualizes SR-IOV resources, making it possible to migrate partitions that rely on the high performance of SR-IOV while still benefiting from Live Partition Mobility.

In summary, Hybrid Network Virtualization on IBM Power servers combines SR-IOV's high-performance I/O with the flexibility of Live Partition Mobility, allowing for improved performance, simplified network configuration, and seamless migration of workloads.

IBM Power Systems firmware level FW950.00 and Hardware Management Console (HMC) version 9.2.950.0, in conjunction with compatible Linux distributions, marked the introduction of Linux Host Network Virtualization (HNV) support. This advancement empowers logical partitions running Linux to leverage the efficiency and performance advantages offered by SR-IOV logical ports while also participating in key mobility operations. These operations include both active and inactive Live Partition Mobility (LPM) and Simplified Remote Restart (SRR).

Enabling HNV is a straightforward process during the configuration of an SR-IOV logical port, where a new “Migratable” option can be selected.

Under the hood, Hybrid Network Virtualization (HNV) utilizes Linux active-backup bonding. This bonding mechanism is crucial for facilitating LPM for virtual partitions that are configured to use SR-IOV logical ports, ensuring a seamless migration process.

Requirements for HNV

The following requirements and conditions must be met to perform the HNV operation:

- Hardware Management Console (HMC) Version 9 Release 2 Maintenance Level 950, or later
- Virtual I/O Server (VIOS) version 3.1.2.0, or later
- Power Hypervisor with firmware at level FW950, or later
- Powerpc-utils version 1.3.8, or later for RHEL 8.4+, SLES15 SP3
- Powerpc-utils version 1.3.10, or later for SLES15 SP4+
- Backend virtual device support
  - IBM virtual Ethernet device (ibmveth)
    - SLES15 SP3, or later
    - RHEL8.4, or later
    - RHEL9.0, or later
  - IBM virtual network interface (ibmvNIC) supported
    - RHEL8.6, or later
    - RHEL9.0 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 paramount in today's digital age. As businesses increasingly rely on technology to operate, protecting sensitive data and preventing cyberattacks becomes a top priority. Guest secure boot with static keys was recently introduced with Linux on IBM Power. This section provides a brief technical overview of this functionality.

Secure boot or verified boot is a firmware and software mechanism that protects the integrity of operating system (OS) boot components. Malicious software such as boot kits and root kits can be used to subvert **`bootloader`** and OS security during the boot process. These threats are countered by public key cryptography. The binary hash of boot components is signed by private keys and their signatures are verified by the corresponding public keys. The boot component image hash is compared with the signature hash and only if they match is the boot process allowed to continue. The verification procedure ensures that the integrity of the boot components is intact.

Linux logical partition (LPAR) secure boot ensures the integrity of the Linux boot stack. The hypervisor and partition firmware are part of the core root of trust. The partition firmware verifies the appended signature on the GRUB image before handing control to GRUB. Similarly, GRUB verifies the appended signature on the kernel image before booting the OS. This ensures that every image that runs at boot time is verified and trusted.

By default, the Linux LPAR secure boot uses static key management. This means that each image (A) embeds the required keys to verify the image (B) that image (A) loads. For example, the keys that are used to verify that the GRUB image are built into the firmware image. Similarly, the keys that are used for verifying that the kernel image are built into the GRUB image. These keys are pre-defined keys and they cannot be modified at run time. Any changes to the static keys require both firmware and OS updates.

Figure 10-2 represents how the static key-based guest secure boot solution works. The numbers represent the chronological order of operation for each individual boot component.

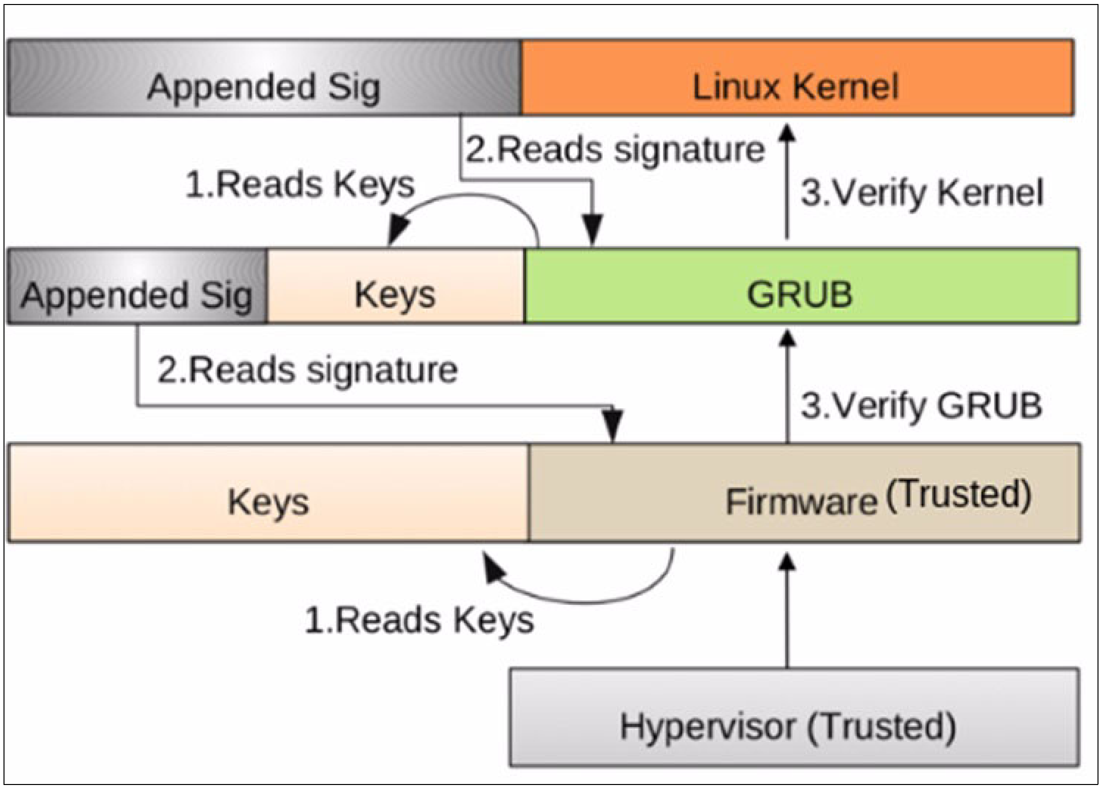


Figure 10-2 Static key based secure boot solution

Secure boot modes

The HMC provides three secure boot modes:

- Disabled
- Enabled and log only
- Enabled and enforced

Linux on Power supports two out 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 additional technical information including supported releases and features refer to https://www.ibm.com/docs/en/linux-on-systems?topic=servers-guest-secure-boot-static-keys

KVM in a PowerVM LPAR

Kernel-based Virtual Machine (KVM) is an extra virtualization option on Power10 systems that run on PowerVM. KVM brings the power, speed, and flexibility of the KVM virtualization technology to a PowerVM logical partition (LPAR). An LPAR that runs a KVM-enabled Linux distribution can host PPC64-LE KVM guests. The KVM Guests can use the existing resources that are assigned to the LPAR.

KVM in a PowerVM LPAR utilizes the industry standard Linux KVM virtualization stack and can easily integrate within an existing Linux virtualization ecosystem. For more information on KVM in Power10, see "Kernel-based Virtual Machine" on page 36.

10.2 Containerization solutions on IBM Power

Modernizing IBM Power infrastructure with containers is a strategic approach for organizations looking to enhance agility, efficiency, and scalability while leveraging their existing investment in robust Power Systems. Containerization provides a way to package applications and their dependencies, enabling faster deployment cycles, improved resource utilization, and a more consistent experience across different environments.

Containerization solutions on IBM Power Systems provide a powerful and flexible way to deploy and manage applications, leveraging the performance and reliability of the Power architecture. This section discusses the various containerization technologies that are available on IBM Power.

10.2.1 OpenShift

OpenShift is an enterprise-grade container orchestration platform developed by Red Hat, built on top of Kubernetes, the open-source system for automating the deployment, scaling, and management of containerized applications. OpenShift simplifies the complexity of managing containers by providing an integrated environment that includes not only container orchestration but also tools for CI/CD (continuous integration and continuous deployment), monitoring, and security. It empowers organizations to build, deploy, and scale applications more efficiently across hybrid and multi-cloud environments.

One of the key features of OpenShift is its developer-centric tools, which streamline the application lifecycle. Developers can quickly create, test, and deploy applications using predefined templates, automated workflows, and integrated development environments. OpenShift provides built-in support for popular programming languages and frameworks, as well as Kubernetes-native resources like pods, services, and persistent storage, which helps developers focus on building their applications rather than managing infrastructure.

In addition to its developer-friendly features, OpenShift offers enterprise-level security and management tools to ensure that containerized applications are secure, scalable, and maintainable. It includes features like integrated role-based access control (RBAC), automated security updates, and centralized logging and monitoring. OpenShift is designed to run in diverse environments, from on-premises data centers to public clouds, making it an ideal solution for organizations seeking to modernize their IT infrastructure and leverage the benefits of containerization and microservices architecture.

OpenShift on IBM Power

Red Hat OpenShift on IBM Power Systems integrates the robust capabilities of IBM's Power architecture with the leading Kubernetes-powered hybrid cloud application platform. This synergy delivers a powerful and adaptable environment for modernizing applications and deploying cloud-native workloads, leveraging OpenShift's comprehensive features for development, deployment, and management alongside IBM Power's renowned performance, reliability, scalability, and security, particularly advantageous for demanding applications like AI, big data, and mission-critical systems. Supporting hybrid cloud deployments, OpenShift on Power enables workload flexibility across on-premises and cloud environments, facilitating application modernization through containerization and the deployment of microservices, all managed efficiently via the Operator Framework and a growing catalog of certified Operators. The benefits include enhanced performance and scalability, superior reliability, optimized resource utilization, robust security, simplified management, potential cost efficiencies, and the ability to leverage existing Power infrastructure, with IBM Cloud Paks further extending the platform's capabilities. Deployment options include User-Provisioned Infrastructure (UPI) for greater control and Installer-Provisioned Infrastructure (IPI) primarily on IBM Cloud Power Systems Virtual Server, with considerations for operating system support, hardware requirements, networking, storage, and licensing. Ultimately, OpenShift on IBM Power offers a strategic combination for organizations seeking a high-performance, reliable, and secure foundation for their modern application initiatives.

IBM Cloud Paks

IBM Cloud Paks are AI-powered software for hybrid cloud that are designed to help you advance digital transformation with prediction, security, automation and modernization capabilities. They let you develop applications once and deploy them anywhere, integrate security across your IT landscape and automate operations with intelligent workflows. Deploy them across any cloud to accelerate development, deliver seamless integration and enhance collaboration and efficiency.

Cloud Paks are designed to help you:

- Modernize with ease

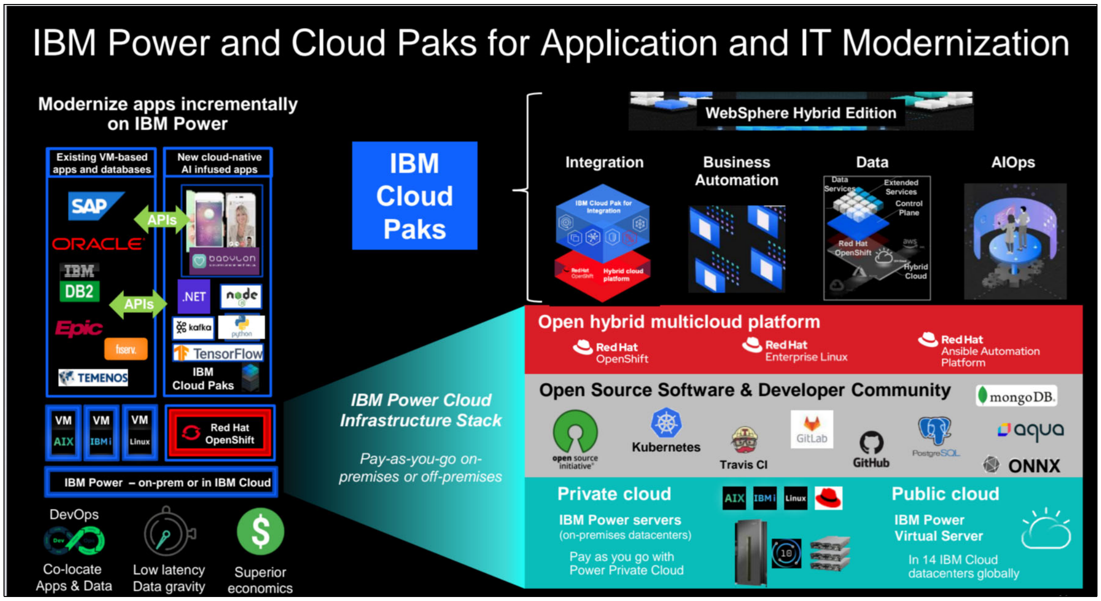
  Develop and consume cloud services anywhere, from any cloud.
- Predict outcomes

  Collect, organize and analyze data regardless of its type or where it lives.
- Automate at scale

  Implement intelligent workflows in your business using AI powered automation.
- Protect your business

  Generate deeper insights into threats and risks across hybrid multicloud environments.

Figure 10-3 shows IBM Cloud Paks on Power.



IBM Power and Cloud Paks for Application and IT Modernization

Modernize apps incrementally on IBM Power

Existing VM-based apps and databases

New cloud-native AI infused apps

APIs

APIs

IBM Cloud Paks

VM AIX

VM IBM i

VM Linux

Red Hat OpenShift

IBM Power – on-prem or in IBM Cloud

DevOps

Co-locate Apps & Data

Low latency Data gravity

Superior economics

IBM Cloud Paks

IBM Power Cloud Infrastructure Stack

Pay-as-you-go on-premises or off-premises

WebSphere Hybrid Edition

Integration

Business Automation

Data

AIOps

Open hybrid multicloud platform

Red Hat OpenShift

Red Hat Enterprise Linux

Red Hat Ansible Automation Platform

Open Source Software & Developer Community

Private cloud

IBM Power servers (on-premises datacenters)

Pay as you go with Power Private Cloud

Public cloud

IBM Power Virtual Server

In 14 IBM Cloud datacenters globally

Figure 10-3 IBM Cloud Paks on IBM Power

More businesses are recognizing that AI is no longer an option. In order to digitally transform and gain a competitive edge, they must embrace AI and start scaling it across their enterprise. This will require building an information architecture that can connect multiple data sources, ensure data quality and fully support their AI data needs. As the value of AI continues to increase, overcoming these challenges will be a priority for organizations.

IBM Cloud Paks are pre-integrated containerized software built on Red Hat OpenShift that are designed to help you develop and consume cloud services anywhere and from any cloud, so you can modernize with ease and make your data work for you, wherever you are. Flexibly and quickly consume and manage all deployments with a governed, protected and unified platform that delivers consistency across software tools and is continuously available – from the data center all the way to the edge.

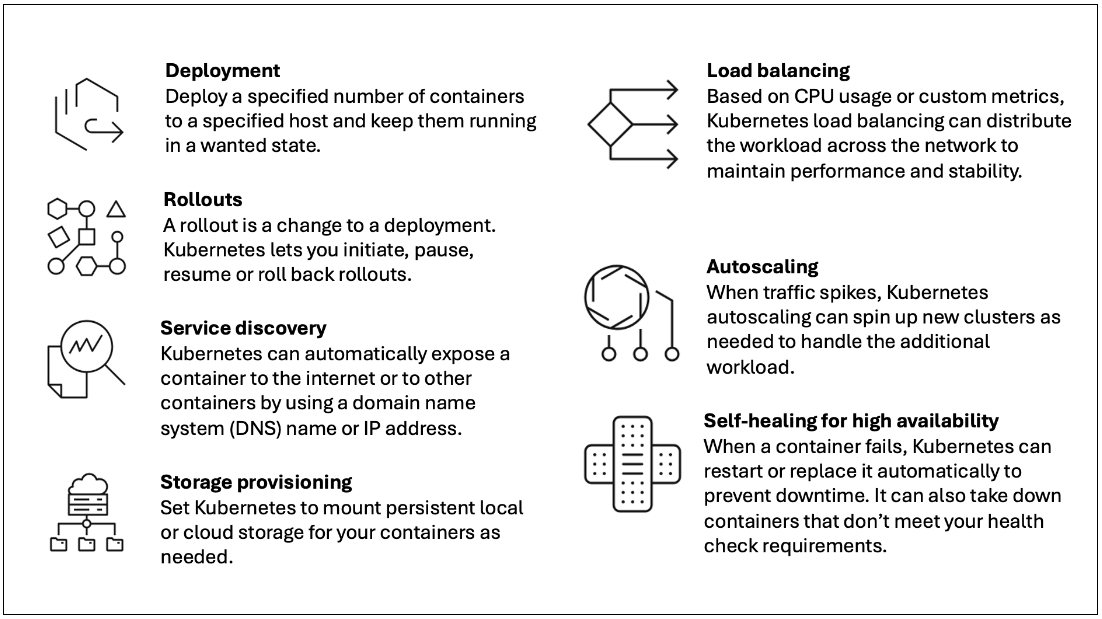
10.2.2 Kubernetes

Kubernetes is an open source container orchestration platform for scheduling and automating the deployment, management and scaling of containerized applications.

Kubernetes enables organizations to deliver a highly productive hybrid multicloud computing environment to perform complex tasks surrounding infrastructure and operations. It also supports cloud-native development by enabling a build-once-and-deploy-anywhere approach to building.

Kubernetes schedules and automates container-related tasks throughout the application lifecycle, including deployment, provisioning, management, and maintenance tasks.

This is shown in Figure 10-4.



Deployment
Deploy a specified number of containers to a specified host and keep them running in a wanted state.

Rollouts
A rollout is a change to a deployment. Kubernetes lets you initiate, pause, resume or roll back rollouts.

Service discovery
Kubernetes can automatically expose a container to the internet or to other containers by using a domain name system (DNS) name or IP address.

Storage provisioning
Set Kubernetes to mount persistent local or cloud storage for your containers as needed.

Load balancing
Based on CPU usage or custom metrics, Kubernetes load balancing can distribute the workload across the network to maintain performance and stability.

Autoscaling
When traffic spikes, Kubernetes autoscaling can spin up new clusters as needed to handle the additional workload.

Self-healing for high availability
When a container fails, Kubernetes can restart or replace it automatically to prevent downtime. It can also take down containers that don't meet your health check requirements.

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. This is shown in Figure 10-5.

A node consists of pods. These are groups of containers that share the same computing resources and the same network. They are also the unit of scalability in Kubernetes. If a container in a pod is gaining more traffic than it can handle, Kubernetes will replicate the pod to other nodes in the cluster.

The control plane automatically handles scheduling the pods across the nodes in a cluster.

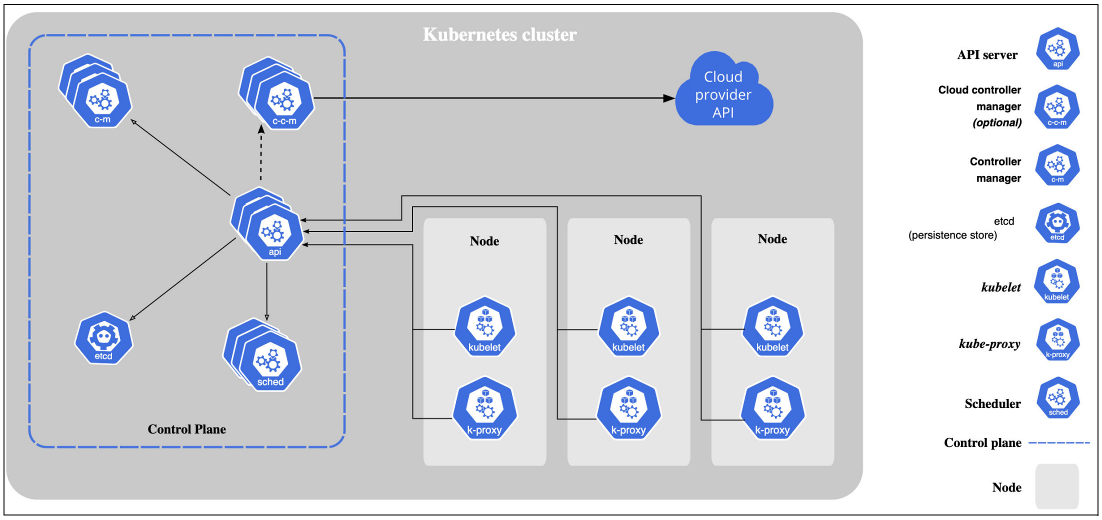


Figure 10-5 Figure - Kubernetes components.

Each cluster has a master node that handles the cluster’s control plane. The master node runs a scheduler service that automates when and where the containers are deployed based on developer-set 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:

- API server

  The application programming interface (API) server in Kubernetes exposes the Kubernetes API (the interface used to manage, create and configure Kubernetes clusters) and serves as the entry point for all commands and queries.

- etcd

  The etcd is an open source, distributed key-value store used to hold and manage the critical information that distributed systems need to keep running. In Kubernetes, etcd manages the configuration data, state data and metadata.

- Scheduler

  This component tracks newly created pods and selects nodes for them to run on. The scheduler considers resource availability and allocation restraints, hardware and software requirements, and more.

- Controller-manager

  A set of built-in controllers, the Kubernetes controller-manager runs a control loop that monitors the shared state of the cluster and communicates with the API server to manage resources, pods or service endpoints. The controller-manager consists of separate processes that are bundled together to reduce complexity and run in one process.

- Cloud-controller-manager

  This component is similar in function to the controller-manager link. It links to a cloud provider’s API and separates the components that interact with that cloud platform from those that only interact within the cluster.

Node Components

Node components run on every node, maintaining running pods and providing the Kubernetes runtime environment:

- kubelet

  Ensures that Pods are running, including their containers.

- kube-proxy (optional)

  Maintains network rules on nodes to implement Services.

- Container runtime

  Software responsible for running containers.

Kubernetes Use Cases

Enterprise organizations use Kubernetes to support the following use cases that all play a crucial role in comprising modern IT infrastructure.

- Microservices architecture or cloud-native development:

  Kubernetes helps ensure that each microservice has the resources it needs to run effectively while also minimizing the operational overhead associated with manually managing multiple containers.

- Hybrid multicloud environments:

  Hybrid cloud combines and unifies public cloud, private cloud and on-premises data center infrastructure to create a singe, flexible, cost-optimized IT infrastructure. Today, hybrid cloud has merged with multicloud, public cloud services from more than one cloud vendor, to create a hybrid multicloud environment. A hybrid multicloud approach creates greater flexibility and reduces an organization's dependency on one vendor, preventing vendor lock-in. Since Kubernetes creates the foundation for cloud-native development, it's key to hybrid multicloud adoption.

- Applications at scale

  Kubernetes supports large-scale cloud app deployment with autoscaling. This process allows applications to scale up or down, adjusting to demand changes automatically, with speed, efficiency and minimal downtime.The elastic scalability of Kubernetes deployment means that resources can be added or removed based on changes in user traffic like flash sales on retail websites.

- Application modernization

  Kubernetes provides the modern cloud platform needed to support application modernization, migrating and transforming monolithic legacy applications into cloud applications built on microservices architecture.

- DevOps practices

  Automation is at the core of DevOps, which speeds the delivery of higher-quality software by combining and automating the work of software development and IT operations teams. Kubernetes helps DevOps teams build and update apps rapidly by automating the configuration and deployment of applications.

- Artificial intelligence (AI) and machine learning (ML)

  The ML models and large language models (LLM) that support AI include components that would be difficult and time-consuming to manage separately. By automating configuration, deployment and scalability across cloud environments, Kubernetes helps provide the agility and flexibility needed to train, test and deploy these complex models.

For more information on 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. Docker is the most widely used containerization tool. Docker is so popular today that “Docker” and “containers” are often used interchangeably.

Containers simplify the development and delivery of distributed applications. They have become increasingly popular as organizations shift to cloud-native development and hybrid multicloud environments. Developers can create containers without Docker by working directly with capabilities built into Linux and other operating systems, but Docker makes containerization faster and easier. Like other containerization technologies, including Kubernetes, Docker plays a crucial role in modern software development, specifically microservices architecture.

Docker architecture

Docker uses a client/server architecture. The following is a breakdown of the core components associated with Docker, along with other Docker terms and tools.

- Docker host

  A Docker host is a physical or virtual machine running Linux (or another Docker-Engine compatible OS).

- Docker Engine

  Docker engine is a client/server application consisting 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

  Docker daemon is a service that creates and manages Docker images, by using the commands from the client. Essentially the Docker daemon serves as the control center for Docker implementation.

- Docker client

  The Docker client provides the CLI that accesses 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 help package and distribute applications. They include images, containers, networks, volumes, plug-ins and more. Docker containers: Docker containers are the live, running instances of Docker images.

  While Docker images are read-only files, containers are live, ephemeral, executable content. Users can interact with them, and administrators can adjust their settings and conditions by using Docker commands.

- Docker images

  Docker images contain executable application source code and all the tools, libraries and dependencies the application code needs to run as a container. When a developer runs the Docker image, it becomes an instance of the container.

  Docker images are made up of layers, and each layer corresponds to a version of the image. Whenever a developer makes changes to an image, a new top layer is created, and this top layer replaces the previous top layer as the current version of the image. Previous layers are saved for rollbacks or to be reused in other projects.

  Each time a container is created from a Docker image, yet another new layer called the container layer is created. Changes made to the container—like adding or deleting files—are saved to the container layer, and these changes only exist while the container is running.

  This iterative image-creation process increases overall efficiency since multiple live container instances can run from a single base image. When they do so, they use a common stack.

- Docker build

  Docker build is a command that has tools and features for building Docker images.

- Dockerfile

  Every Docker container starts with a simple text file containing instructions for how to build the Docker container image. Dockerfile automates the process of creating Docker images. It's essentially a list of CLI instructions that Docker Engine will run to assemble the image. The list of Docker commands is vast but standardized: Docker operations work the same regardless of contents, infrastructure or other environment variables.

- Docker Hub

  Docker Hub is the public repository of Docker images, calling itself the world's largest library and community for container images. Docker Hub includes images produced by Docker, Inc., certified images belonging to the Docker Trusted Registry and thousands of other images.

- Docker Desktop

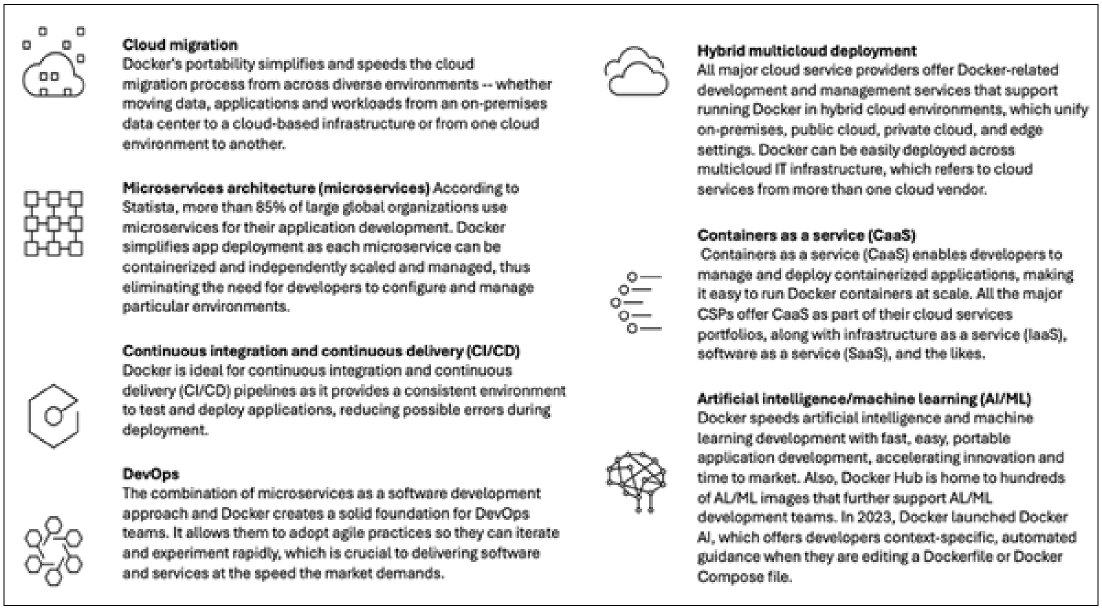
  Docker Desktop is an application for Mac or Windows that includes Docker Engine, Docker CLI client, Docker Compose, Kubernetes and others. It also provides access to Docker Hub.

- Docker registry

  A Docker registry is a scalable, open-source storage and distribution system for Docker images. It enables developers to track image versions in repositories by using tagging 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 available for businesses.



Cloud migration
Docker's portability simplifies and speeds the cloud migration process from across diverse environments -- whether moving data, applications and workloads from an on-premises data center to a cloud-based infrastructure or from one cloud environment to another.

Microservices architecture (microservices) According to Statista, more than 85% of large global organizations use microservices for their application development. Docker simplifies app deployment as each microservice can be containerized and independently scaled and managed, thus eliminating the need for developers to configure and manage particular environments.

Continuous integration and continuous delivery (CI/CD)
Docker is ideal for continuous integration and continuous delivery (CI/CD) pipelines as it provides a consistent environment to test and deploy applications, reducing possible errors during deployment.

DevOps
The combination of microservices as a software development approach and Docker creates a solid foundation for DevOps teams. It allows them to adopt agile practices so they can iterate and experiment rapidly, which is crucial to delivering software and services at the speed the market demands.

Hybrid multicloud deployment
All major cloud service providers offer Docker-related development and management services that support running Docker in hybrid cloud environments, which unify on-premises, public cloud, private cloud, and edge settings. Docker can be easily deployed across multicloud IT infrastructure, which refers to cloud services from more than one cloud vendor.

Containers as a service (CaaS)
Containers as a service (CaaS) enables developers to manage and deploy containerized applications, making it easy to run Docker containers at scale. All the major CSPs offer CaaS as part of their cloud services portfolios, along with infrastructure as a service (IaaS), software as a service (SaaS), and the likes.

Artificial intelligence/machine learning (AI/ML)
Docker speeds artificial intelligence and machine learning development with fast, easy, portable application development, accelerating innovation and time to market. Also, Docker Hub is home to hundreds of AL/ML images that further support AL/ML development teams. In 2023, Docker launched Docker AI, which offers developers context-specific, automated guidance when they are editing a Dockerfile or Docker Compose file.

Figure 10-6 Docker use cases

Docker announced new AI functions in 2023[1]. Docker AI, Docker's first AI-powered product, is aimed at boosting developer productivity by tapping into the universe of Docker developer wisdom to provide context-specific, automated guidance to developers as they work.

Docker AI provides context-specific, automated guidance to developers when they are editing a Dockerfile or Docker Compose file, debugging their local 'docker build,' or running a test locally. Docker AI enables developers to benefit from the collective wisdom of the millions of developers using Docker – some for more than 10 years – through automatically generating best practices and selecting up-to-date, secure images for their applications. Using Docker AI, developers are able to spend more time focused on their app, less on tools and infrastructure.

[1] https://www.docker.com/press-release/announces-ai-boosting-developer-productivity-through-automated-guidance/#

10.2.4 Podman

Podman (short for pod manager) is an open source tool for developing, managing, and running containers. Developed by Red Hat engineers along with the open source community, Podman manages the entire container ecosystem using the libpod library.

Podman’s daemonless and inclusive architecture makes it an accessible, security-focused option for container management. Its accompanying tools and features, such as Buildah and Skopeo, let developers customize their container environments to suit their needs. Developers can also take advantage of Podman Desktop, a graphical user interface (GUI) for using Podman in local environments. Users can run Podman on various Linux distributions, such as Red Hat Enterprise Linux, Fedora, CentOS, and Ubuntu.

Podman stands out from other container engines because it’s daemonless, meaning it doesn't rely on a process with root privileges to run containers. Daemons are processes that run in the background of your system to do the work of running containers without a user interface. Podman cuts out the daemon and lets regular users run containers without interacting with a root-owned daemon, or allows for the use of rootless containers. By going rootless, users can create, run, and manage containers without requiring processes with admin privileges, making your container environment more accessible while reducing security risks. Additionally, Podman launches each container with a Security-Enhanced Linux (SELinux) label, giving administrators more control over what resources and capabilities are provided to container processes.

Users can invoke Podman from the command line to pull containers from a repository and run them. Podman calls the configured container runtime to create the running container. But without a dedicated daemon, Podman uses systemd – a system and service manager for Linux operating systems – to make updates and keep containers running in the background. By integrating systemd and Podman, you can generate control units for your containers and run them with systemd automatically enabled.

Podman also deploys a RESTful API (REST API) to manage containers. REST stands for representational state transfer. A REST API is an API that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. With the REST API, you can call Podman from platforms such as cURL, Postman, Google's Advanced REST client, and many others.

Podman offers the same high-performance capabilities as leading container engines, but with the flexibility, accessibility, and security features that many development teams are seeking. Podman can help you:

- Manage container images and the full container lifecycle, including running, networking, checkpointing, and removing containers.
- Run and isolate resources for rootless containers and pods.
- Support OCI and Docker images as well as 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 functionality.
- Implement checkpoint/restore functionality 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 if an updated container fails to start and automatically rolls back to the last working version. This provides new levels of reliability for applications.

For more information on Podman see What is Podman?

Part 3

Appendices

In this part we provide additional details that will be useful as you work to modernize your IBM Power infrastructure and applications.

Appendix A, “Components used in modernization on IBM Power” on page 343 is a list of components and tools that can be utilized as you modernize your IBM Power infrastructure. This is provided to give you a feel for the number of options available to provide a more modern infrastructure and to modernize your applications.

Appendix B, “Modernization using cloud native tools” on page 387 provides information on utilizing cloud-native tools on Skytap on Azure. It discusses key Azure-native services that enable integration, improve scalability, and enhance automation for IBM i, AIX, and Linux for POWER environments.

Appendix C, “Details for Performance Claims” on page 407 provides the details supporting the performance and financial claims made in the book.

© Copyright IBM Corp. 2025.

A

Components used in modernization on IBM Power

This publication has detailed various approaches to modernizing your applications and workloads on the robust IBM Power platform, ultimately increasing the value it delivers to your business. While IBM Power is recognized for its reliability and high performance in running critical applications, we've shown how integrating modern digital interfaces and AI can further enhance its capabilities and generate significant business advantages.

This appendix outlines several components that can help in modernizing your IBM Power infrastructure. While it is not an exhaustive list, it presents a selection of tools organized into broad categories.

The following categories of components are covered:

- A.1, “Red Hat” on page 344
- A.2, “IBM” on page 359
- A.3, “Open Source” on page 364
- A.4, “Independent Software Vendors” on page 375

© Copyright IBM Corp. 2025.

A.1 Red Hat

Red Hat plays a pivotal role in the modernization of IT infrastructure and applications, helping organizations transition from legacy systems to modern, cloud-native environments. By leveraging Red Hat OpenShift, businesses can replatform existing workloads onto a Kubernetes-based platform, breaking down monolithic applications into microservices. This transformation not only enhances performance and scalability but also improves security and compliance. Red Hat's modernization approach includes adopting DevOps practices, continuous integration/continuous deployment (CI/CD), and site reliability engineering (SRE), which streamline workflows and boost engineering productivity.

Modernization with Red Hat also brings significant cost savings and operational efficiencies. By containerizing applications and utilizing OpenShift's self-healing and auto-scaling capabilities, organizations can optimize resource usage and reduce maintenance expenses. Additionally, Red Hat's tools and frameworks support hybrid and multi-cloud deployments, providing flexibility and ensuring that applications can run seamlessly across different environments. This comprehensive approach to modernization helps businesses stay competitive, improve user experiences, and achieve their digital transformation goals.

A.1.1 OpenShift

Red Hat OpenShift offers numerous benefits, including a robust and scalable platform for developing, deploying, and managing containerized applications. It provides a consistent hybrid cloud foundation, enabling seamless operations across on-premises, public cloud, and edge environments. OpenShift enhances developer productivity with integrated developer tools, CI/CD pipelines, and automated workflows. Its enterprise-grade security features, such as built-in compliance and automated updates, ensure applications remain secure and up-to-date. Additionally, OpenShift's support for multi-architecture clusters and Single Node OpenShift (SNO) allows organizations to optimize their infrastructure for diverse workloads and resource-constrained environments, making it a versatile solution for modern IT needs.

A.1.1.1 Recent support

Red Hat OpenShift has recently introduced two significant functions: Single Node OpenShift (SNO) and multi-architecture clusters to their product portfolio. Both of these functions provide some significant advantages in the IBM Power environment and are discussed in this section.

Single Node OpenShift

The standard OpenShift implementation typically involves multiple control plane nodes and worker nodes to ensure high availability. However, for environments that do not require hardware redundancy and aim to conserve resources, Single Node OpenShift (SNO) is an excellent option. SNO is a specialized deployment of Red Hat OpenShift that consolidates both control plane and worker node functions into a single server. This setup is particularly beneficial for edge computing environments where space, power, and connectivity are limited. Designed to operate autonomously, SNO is ideal for scenarios such as intermittent connectivity, portable clouds, and 5G radio access networks (RAN) near base stations. By utilizing a single node, organizations can deploy a full OpenShift environment with fewer resources, simplifying the setup and management of Kubernetes clusters in constrained environments.

The benefits of Single Node OpenShift include reduced resource requirements and simplified management. By combining control and worker node capabilities, SNO provides a consistent OpenShift experience across various deployment sizes, from large data centers to remote edge locations. This consistency allows organizations to leverage the same tools and skills across their entire infrastructure, ensuring seamless operations and upgrades. Additionally, SNO's ability to run autonomously makes it suitable for environments with limited or unreliable connectivity, enabling continuous operations even in challenging conditions. Overall, Single Node OpenShift offers a flexible and efficient solution for deploying Kubernetes in resource-constrained environments.

Multiple Architecture Clusters

The release of Red Hat OpenShift 4.14 brought the OpenShift Container Platform Multiple-Architecture Compute feature to IBM Power. Multi-Arch Compute provides a single heterogeneous cluster, enabling fit-for-purpose computing so clients can align tasks and applications to CPU strengths and software availability rather than one architecture. This support was expanded in Red Hat OpenShift 4.15 which enabled a Red Hat OpenShift cluster to support an IBM Power control plane and add x86 architecture worker nodes.

Multi-Arch Compute for OpenShift Container Platform lets you use a pair of compute architectures, such as ppc64le and amd64, within a single cluster. This exciting feature opens new possibilities for versatility and optimization for composite solutions that span multiple architectures.

For more information on Multiple Architecture Clusters on IBM Power refer to *Creating OpenShift Multiple Architecture Clusters with IBM Power*, SG24-8565

A.1.1.2 OpenShift Add-ons

Red Hat OpenShift offers a variety of add-ons that enhance the functionality and capabilities of your OpenShift clusters. These add-ons include services like monitoring, logging, security, and networking, which can be easily integrated using the OpenShift Cluster Manager. For example, you can add the cluster-logging-operator to enable comprehensive logging capabilities or integrate with external monitoring tools like Amazon CloudWatch. These add-ons help streamline operations, improve visibility, and ensure that your OpenShift environment is optimized for performance and reliability. By leveraging these add-ons, organizations can tailor their OpenShift deployments to meet specific needs 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 leverages the Tekton framework. OpenShift Pipelines allows developers to create advanced CI workflows for their applications, automating and speeding up the delivery process. Each step of the CI/CD pipeline runs in its own container, enabling independent scaling to meet the demands of the pipeline. This serverless approach eliminates the need for managing a central CI/CD server, providing full control over delivery pipelines, plugins, and access control. OpenShift Pipelines integrates seamlessly with the OpenShift console, allowing developers to configure and execute pipelines directly alongside their applications.

Tekton, the underlying framework for OpenShift Pipelines, is an open-source project that provides Kubernetes-native CI/CD components. Tekton's flexible and powerful architecture supports standard Kubernetes-native pipeline definitions, extensibility to build images using various Kubernetes tools, and portability across any Kubernetes distribution. Tekton also offers a robust CLI for interacting with pipelines and an integrated user experience within the OpenShift web console. By leveraging Tekton, OpenShift Pipelines provides a streamlined and scalable CI/CD experience, enabling organizations to 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 designed for Kubernetes. It automates the deployment and lifecycle management of applications by using Git repositories as the source of truth for defining the desired application state. This approach ensures that application definitions, configurations, and environments are version-controlled and auditable. Argo CD continuously monitors running applications, comparing the live state against the desired target state specified in the Git repository. If discrepancies are found, Argo CD can automatically or manually sync the live state back to the desired state, ensuring consistency and reliability.

One of the key features of Argo CD is its support for multiple configuration management and templating tools, such as Kustomize, Helm, and Jsonnet. It also offers robust security features, including single sign-on (SSO) integration with various identity providers and role-based access control (RBAC) policies for authorization. Additionally, 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 deployment process, Argo CD helps organizations achieve faster, more reliable software delivery.

Red Hat OpenShift Serverless

Red Hat OpenShift Serverless, simplifies the development and deployment of cloud-native applications by abstracting the underlying infrastructure complexities. Based on the open-source Knative project, OpenShift Serverless allows developers to focus on writing code without worrying about managing servers, scaling, or provisioning resources. This approach enhances developer productivity by automatically scaling applications up or down based on demand, including scaling to zero when there are no active requests. This ensures efficient resource utilization and cost savings, as you only pay for the resources you use.

OpenShift Serverless integrates seamlessly with other OpenShift services, such as Service Mesh, Pipelines, and GitOps, providing a comprehensive and efficient development and deployment experience. It supports hybrid and multi-cloud environments, ensuring that applications are portable and consistent across different platforms. Additionally, OpenShift Serverless includes built-in security features, such as integrated security monitoring and automation, to ensure that applications are secure, reliable, and highly available. By leveraging OpenShift Serverless, organizations can accelerate their time-to-market for new services and features while optimizing resource utilization and reducing operational costs.

Odo

Odo is an open-source, developer-focused tool designed to simplify the development of applications for Kubernetes and OpenShift. It provides a fast and iterative development experience by allowing developers to build, test, and deploy applications directly from their local machine to a Kubernetes cluster. Odo abstracts the complexities of Kubernetes, making it easier for developers to focus on writing code without worrying about the underlying infrastructure.

Odo offers several key features designed to simplify and enhance the development experience for Kubernetes and OpenShift environments:

- Fast and Iterative Development: Odo allows developers to quickly build, test, and deploy applications directly from their local machine to a Kubernetes cluster, streamlining the development workflow.
- Language and Framework Support: It supports multiple languages and frameworks, making it versatile for various development needs.
- Integration with IDEs: Odo integrates seamlessly with popular Integrated Development Environments (IDEs), enhancing the developer experience.

- Simple Command-Line Interface: The tool provides a straightforward command-line interface for managing application components and services.
- Deployment Flexibility: Odo supports on-premises, cloud, and hybrid cloud environments, offering flexibility in deployment options.

These features make Odo a powerful tool for developers looking to efficiently manage their Kubernetes and OpenShift applications.

Dev Spaces

Red Hat OpenShift Dev Spaces is a Kubernetes-native development solution that provides cloud-based development environments for enterprise teams. Built on the open-source Eclipse Che project, OpenShift Dev Spaces uses Kubernetes and containers to offer consistent, secure, and zero-configuration development environments. These environments are accessible through a web browser, allowing developers to code, build, test, and run applications on OpenShift without worrying about the complexities of Kubernetes management.

Key features of OpenShift Dev Spaces include integration with popular IDEs like Visual Studio Code and JetBrains IntelliJ IDEA, support for defining development environments as code using the devfile format, and robust security measures such as OpenShift OAuth and LDAP or Active Directory integration. This tool helps streamline the development workflow by providing a fast and familiar experience similar to a local IDE, but with the added benefits of cloud-based scalability and centralized management.

Cert-manager Operator

The cert-manager Operator is a Kubernetes-native tool designed to automate the management of certificates within a cluster. It integrates seamlessly with Red Hat OpenShift, providing a cluster-wide service for certificate lifecycle management. The cert-manager Operator allows you to integrate with external certificate authorities and provides 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 offers tools for automatic certificate renewal and self-service certificate management. This operator simplifies the process of securing applications and services within a Kubernetes cluster, ensuring that certificates are always up-to-date and properly managed.

Cost Management Metrics Operator

The Cost Management Metrics Operator is a tool designed to gather and analyze data from OpenShift Container Platform to help organizations manage and optimize their costs. It integrates seamlessly with Red Hat OpenShift and provides detailed insights into resource usage, enabling better forecasting and capacity planning. The operator collects metrics such as CPU and memory usage, requests, and limits, and can capture up to 90 days of historical data. This historical data helps identify trends and patterns, allowing organizations to make informed decisions about their cluster's capacity needs and optimize costs accordingly.

One of the key benefits of the Cost Management Metrics Operator is its ability to automate the collection and analysis of cost-related metrics, reducing the manual effort 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. Additionally, it supports restricted network modes and can integrate 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 their IT costs, ensuring they get the maximum return on their investment in technology.

A.1.1.3 Additional Red Hat OpenShift Products

Red Hat provides additional software components or tools that enhance and extend the capabilities of Red Hat OpenShift. These add-ons are typically designed to address specific needs, including improved security, application management, cloud integration, monitoring, and storage management and are charged separately. By incorporating these add-ons into their existing Red Hat infrastructure, organizations can streamline operations, improve performance, and leverage additional features to optimize their workflows. This section discusses some of these add-on products.

Red Hat Quay

Red Hat Quay is a secure and highly available container image registry platform. Red Hat Quay is a self-managed product that works with most of the orchestration systems and container environments. Additionally, Red Hat Quay.io is offered as a hosted SaaS solution.

With Quay self-managed service, you can run the registry in an offline environment locally, mirror, replicate and integrate repository with Red Hat OpenShift. Quay fully managed service will provide a global content network for serving images, Helps generate vulnerability reports for the images, CI/CD integration to pipelines and provides possibility to build container images based on source code commits via GitHub or GitLab

Red Hat Quay - Self managed

The latest version of Red Hat Quay 3.13 can be installed by following the documentation at - https://docs.redhat.com/en/documentation/red_hat_quay/3.13. The operator is available on the Operator Hub on Red Hat OpenShift. After successful installation of the operator, the repository could be used to push and pull images as a self-managed service

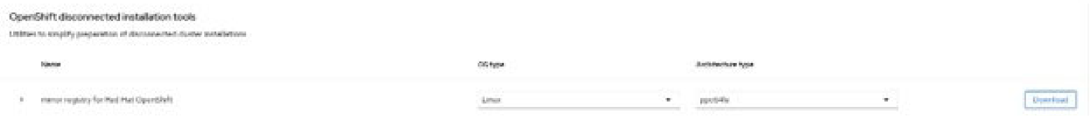
Red Hat Quay - Managed service

The managed service is hosted at https://quay.io/repository/ where a user can register and create a repository. Once the repository is created, Images can be pushed and pulled from the repository. The pricing is based on the number of private repositories per registry account

Red Hat Mirror-Registry-EE

A cut down version of Quay is provided as the mirror registry for Red Hat OpenShift. This is a small and streamlined container registry that you can use as a target for mirroring the required container images of OpenShift Container Platform for disconnected installations.

You will need an OpenShift Container Platform subscription to enable you to download the mirror registry installation files which is also available for ppc64le Linux on Power. This is shown in Figure A-1.



OpenShift disconnected installation tools

Utilities to simplify preparation of disconnected cluster installations

Name	OS type	Architecture type	
mirror registry for Red Hat OpenShift	Linux ▾	ppc64le ▾	Download

Figure A-1

The mirror registry is distributed by a mirror registry execution environment image which provides the cut down quay registry for you to upload the required OpenShift images for installation.

Note: Please visit Red Hat for more information on how to use the mirror-registry for an offline OpenShift installation.

https://docs.openshift.com/container-platform/4.16/installing/disconnected_install/installing-mirroring-creating-registry.html

Red Hat Advanced Cluster Security for Kubernetes

Red Hat Advanced Cluster Security for Kubernetes (RHACS) is a Kubernetes-native security platform that enables organizations to more effectively secure their containerized applications and Kubernetes infrastructure. It helps to secure 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 the Kubernetes infrastructure by offering Kubernetes Security Posture Management (KSPM) capabilities, hardening the underlying infrastructure and 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 to 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 powerful tool designed to help organizations manage and govern their Kubernetes environments at scale. Essentially, it provides a centralized platform for managing multiple Kubernetes clusters, whether they're deployed on-premises, in public clouds, or at the edge.

RHACM allows you to manage numerous Kubernetes clusters from a single control plane, simplifying operations and providing a unified view of your entire Kubernetes infrastructure. It facilitates the provisioning, upgrading, and decommissioning of clusters, streamlining cluster lifecycle management. RHACM enables you to define and enforce policies across your clusters, ensuring compliance with security and regulatory requirements, which helps maintain consistency and reduces the risk of configuration drift. RHACM simplifies the deployment and management of applications across multiple clusters by providing tools for deploying, updating, and rolling back applications, making it easier to manage complex deployments. Further, it provides the ability to enforce security policies and to check for compliance across all managed clusters, giving a centralized view of security posture.

In essence, RHACM addresses the challenges of managing Kubernetes in complex, distributed environments. It helps organizations to increase operational efficiency, improve security and compliance, and accelerate application delivery. Therefore, RHACM is a very useful tool for organizations as they expand the number of Kubernetes clusters in their environment.

Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation is a comprehensive persistent storage and cluster data management solution designed to integrate seamlessly with and optimize Red Hat OpenShift. It offers a distributed, scalable software-defined storage platform that provides advanced enterprise-level cluster data management services, enabling applications to interact with data in a consistent, simplified, and scalable manner. With multicloud data management capabilities, OpenShift Data Foundation empowers organizations to extend and federate data across multiple infrastructures.

As a robust data foundation for modern production workloads and applications, OpenShift Data Foundation can run anywhere Red Hat OpenShift is deployed—whether on-premise, in public or private cloud environments, or at the edge. The platform delivers agile, flexible data access using standard protocols such as file, block, and object storage, making it ideal for a wide range of workloads and applications. It abstracts the complexities and inconsistencies of different underlying storage infrastructures, all while providing the sophisticated data management services that organizations need.

Designed specifically for container-based environments, OpenShift Data Foundation also supports Red Hat OpenShift Virtualization, providing a unified approach to managing both containers and virtual machines. With the help of a supported Red Hat OpenShift operator, the platform is simple to install and manage, becoming an integral part of the container-based application lifecycle, including cloud-native container management, scheduling, and orchestration.

A.1.2 OpenShift Platform Plus

Red Hat OpenShift Platform Plus is an integrated platform designed to help you build, modernize, and scale applications. With features like multicluster security, compliance, and comprehensive application and data management, it ensures consistency across different infrastructures throughout the software supply chain. By providing a complete suite of services, Red Hat OpenShift Platform Plus enables faster and smarter development, helping you bring applications to market 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 robust and consistent hybrid cloud foundation, empowering organizations to build and scale critical containerized applications with confidence. With a proven track record of supporting business-critical workloads for thousands of customers globally, OpenShift facilitates seamless cloud transitions and enables the creation of cutting-edge customer applications. As a leading contributor to Kubernetes, Red Hat ensures OpenShift remains at the forefront of container orchestration.

- Red Hat Advanced Cluster Management for Kubernetes

  Red Hat Advanced Cluster Management for Kubernetes (RHACM) is a powerful tool designed to help organizations manage and govern their Kubernetes environments at scale. Essentially, it provides a centralized platform for managing multiple Kubernetes clusters, whether they're deployed on-premises, in public clouds, or at the edge. More detail is provided in “Red Hat Advanced Cluster Management for Kubernetes” on page 349.

- 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 details see “Red Hat Advanced Cluster Security for Kubernetes” on page 349

- Red Hat OpenShift Data Foundation

  Red Hat OpenShift Data Foundation is a comprehensive persistent storage and cluster data management solution for Red Hat OpenShift. It offers a distributed, scalable software-defined storage platform that provides advanced enterprise-level cluster data management services. For more details see “Red Hat OpenShift Data Foundation” on page 349.

- Red Hat Quay

  Red Hat Quay, a container image registry service is included as a component of Red Hat OpenShift Platform Plus. Quay is described in “Red Hat Quay” on page 348

A.1.3 Middleware & Application Services

A.1.3.1 AMQ Streams

AMQ Streams is a powerful data streaming platform developed by Red Hat, leveraging Apache Kafka and Apache ZooKeeper. It is designed to handle large-scale, high-performance data streaming and processing tasks. One of its key benefits is scalability, allowing it to manage increasing loads by adding more nodes to the cluster. Additionally, AMQ Streams ensures high data throughput and low latency, making it suitable for real-time data processing. Reliability is another significant advantage, with features like data replication and fault tolerance ensuring data availability and consistency. The platform supports various messaging patterns and integrates seamlessly with other applications, endpoints, and devices, enhancing its versatility. Furthermore, AMQ Streams accommodates multiple programming languages and protocols, making it adaptable for different use cases. Its event-driven architecture is ideal for applications that need to respond to data changes in real-time, providing a robust solution for modern data-driven environments.

A.1.3.2 Fuse

Red Hat Fuse is an open-source integration platform that simplifies connecting applications, data, and devices across hybrid environments. Built on Apache Camel, it provides a robust framework for developing, deploying, and managing integration solutions. Fuse supports various deployment options, including on-premises, cloud, and hybrid environments, making it versatile for different infrastructure needs. Its cloud-native capabilities enable seamless integration with modern platforms like Kubernetes and Red Hat OpenShift, enhancing scalability and flexibility.

One of the key strengths of Red Hat Fuse is its extensive library of connectors and integration patterns, which streamline the process of connecting disparate systems. This feature, combined with its support for microservices architecture, allows organizations to break down monolithic applications into more manageable and scalable components. Fuse also promotes agile development practices, facilitating faster iteration and deployment of integration solutions. By fostering collaboration among integration experts, developers, and business users, Red Hat Fuse ensures that modernization efforts are aligned with business goals and technical requirements.

A.1.3.3 3scale API Management

Red Hat 3scale API Management is a comprehensive platform designed to manage APIs for both internal and external users. It allows organizations to share, secure, distribute, control, and monetize their APIs on a robust infrastructure built for performance, customer control, and future growth. The platform features a unique hybrid-cloud architecture, separating API management policy execution (traffic managers or API gateways) from API management policy configuration (API manager). This separation ensures that API calls do not need to be routed through the 3scale manager infrastructure, enhancing performance with minimal latency. Additionally, 3scale provides a rich API admin portal, including performance dashboards and developer-facing portals for exposing and documenting APIs.

The benefits of using 3scale API Management are numerous. It offers flexible scalability, allowing organizations to add new gateways and scale horizontally with their load. The platform also reduces operational costs through automated signup and billing processes. Furthermore, 3scale accelerates time to market by simplifying the build-and-deploy process for rolling out new features. It includes a custom developer portal and interactive API documentation to help developers get started quickly. The platform's ability to manage APIs efficiently and securely makes it an ideal choice for businesses looking to leverage their APIs for growth and innovation.

A.1.3.4 GitOps

GitOps is an operational framework that applies DevOps best practices to infrastructure automation, using Git as the single source of truth. It leverages Git repositories to manage and control infrastructure through code, enabling automated deployments and seamless rollbacks. By utilizing Git pull requests, GitOps integrates changes automatically, ensuring consistency and reliability in the deployment process. This approach simplifies infrastructure management, reduces manual intervention, and enhances collaboration among development and operations teams.

One of the key features of GitOps is its ability to provide a declarative infrastructure, where the desired state of the system is defined in Git repositories. This ensures that any changes to the infrastructure are tracked and versioned, allowing for easy auditing and compliance. GitOps also enhances security by using Git's version control system to manage access and changes, reducing the attack surface and enabling quick recovery in case of incidents. The use of CI/CD pipelines in GitOps workflows further automates the deployment process, improving efficiency and reducing downtime.

GitOps has many benefits. It improves productivity by automating repetitive tasks and allowing developers to focus on more strategic work. GitOps enhances collaboration by providing a transparent and consistent workflow, where all changes go through the same review and approval process. This fosters better communication and teamwork, as ideas and feedback can be shared easily. Additionally, GitOps reduces operational costs by optimizing resource management and minimizing manual oversight. Overall, GitOps offers a robust and efficient approach to managing cloud-native applications and infrastructure, making it a valuable tool for modern development practices.

A.1.3.5 Ansible Automation Platform

The Red Hat Ansible Automation Platform is a comprehensive solution designed to streamline and scale automation across various domains. It provides everything needed to create, execute, and manage automation tasks, all within a single subscription. The platform includes tools like Ansible Tower, which offers a centralized interface for managing and monitoring automation workflows, and Ansible Galaxy, which provides access to per-built automation roles and modules. Additionally, Ansible Automation Hub offers certified content collections from Red Hat and its partners, ensuring reliable and secure automation.

One of the key benefits of the Ansible Automation Platform is its ability to enhance efficiency and reduce complexity in IT operations. By automating repetitive tasks, it frees up valuable time for IT teams to focus on strategic initiatives. The platform also supports event-driven automation, allowing for real-time responses to changes in the IT environment. Furthermore, it integrates seamlessly with existing IT infrastructure, providing flexibility and scalability to meet evolving business needs. With features like automation analytics and Red Hat Insights, organizations can gain deeper insights into their automation processes and optimize performance.

A.1.4 Runtimes

This section describes several Runtimes that are often used in modernization projects.

A.1.4.1 JBoss

JBoss is an open-source application server 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 EE (Enterprise Edition) standards.
- Modular Architecture: It has a modular design, allowing developers to use only the components they need, which helps in optimizing performance and resource usage.
- Integration: JBoss integrates well with various development tools and frameworks, making it easier to develop and manage applications.
- 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, ensuring 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. JBoss plays a significant role in the modernization of IBM Power systems by enabling the transition 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 seamlessly with OpenShift on IBM Power servers. This allows for the deployment of cloud-native applications and the modernization of existing applications without changing the underlying hardware 1.
- Microservices Architecture: JBoss supports the development of microservices, which can be deployed on IBM Power systems. This architecture enhances scalability, flexibility, and ease of management, making it easier to adapt to dynamic customer demands 1.
- Containerization: JBoss applications can be containerized using platforms like OpenShift. This containerization facilitates the efficient management and orchestration of applications, leading to improved resource utilization and faster deployment cycles 2.
- Enhanced Performance and Security: By leveraging JBoss on IBM Power systems, organizations can benefit from improved performance and security features. IBM Power servers offer workload isolation and platform integrity, ensuring that applications run securely and efficiently 1.
- Cost Savings and Efficiency: Running JBoss on IBM Power servers can lead to significant cost savings. The combination of JBoss and IBM Power helps automate and manage infrastructure tasks, allowing teams to focus on higher-value projects 1.

Overall, JBoss, in conjunction with IBM Power systems, provides a robust framework for modernizing applications, enhancing performance, and achieving greater operational efficiency.

A.1.4.2 JBoss Enterprise Application Platform

JBoss Server, specifically the JBoss Enterprise Application Platform (JBoss EAP), is a powerful and flexible application server used for building, deploying, and hosting Java-based applications. Here are some key features of JBoss EAP:

- Modular Architecture: JBoss EAP is designed with a modular architecture, allowing developers to use only the components they need. This helps optimize performance and resource usage.
- Java EE Compatibility: It supports Java EE standards, making it suitable for enterprise-level applications. This includes support for Enterprise Java Beans (EJB), JavaServer Faces (JSF), and Java Persistence API (JPA).
- High Availability and Clustering: JBoss EAP provides features for high availability and clustering, ensuring that applications can scale and remain resilient under heavy loads.
- Management and Monitoring: It includes tools for managing and monitoring applications, such as a management command line interface (CLI) and web-based management console.
- Security: JBoss EAP offers robust security features, including authentication, authorization, and encryption.
- Integration: It integrates well with various development tools and frameworks, enhancing the development and deployment process.

A.1.4.3 Quarkus

Quarkus is a powerful modernization tool, offering a range of benefits for enterprises looking to update their legacy applications. As a full-stack, cloud-native Java framework developed by Red Hat, Quarkus is designed to optimize Java applications for containerized environments. It supports both Java Virtual Machine (JVM) and native compilation, enabling faster startup times and lower memory usage. This makes Quarkus ideal for modernizing applications to run efficiently in cloud and Kubernetes environments. By leveraging Quarkus, organizations can transform their traditional Java EE applications into lightweight, high-performance microservices, enhancing scalability and reducing operational costs.

Additionally, Quarkus integrates seamlessly with IBM Power systems, providing robust support for hybrid cloud deployments. This integration allows enterprises to take advantage of IBM Power's high performance and reliability while modernizing their application stack. Quarkus also supports various enterprise features such as RESTful APIs, configuration management, service invocation, resilience, security, and monitoring, making it a comprehensive solution for application modernization. By adopting Quarkus, businesses can ensure their applications are future-ready, capable of meeting the demands of modern IT environments.

A.1.4.4 AMQ Broker

AMQ, or Red Hat AMQ, is a suite of messaging tools designed to facilitate reliable and scalable communication between applications. It includes several components, such as AMQ Broker, AMQ Streams, and AMQ Online, each serving different purposes:

- AMQ Broker: Based on Apache ActiveMQ, this component provides robust messaging capabilities, supporting various protocols like MQTT, AMQP, and STOMP. It ensures reliable message delivery and supports high availability and clustering.
- AMQ Streams: This component is based on Apache Kafka and offers a distributed, high-performance data streaming platform that enables real-time data processing and integration.
- AMQ Online: Provides a cloud-native messaging service, allowing users to deploy and manage messaging infrastructure in Kubernetes environments.

AMQ plays a significant role in modernizing applications. It integrates seamlessly with IBM Power, enhancing the ability to build modern, scalable, and resilient applications. For instance, AMQ Streams can be deployed on Red Hat OpenShift Container Platform on IBM Power, enabling high-performance data streaming and real-time processing. This integration allows enterprises to leverage IBM Power's high performance and reliability while modernizing their messaging infrastructure, ensuring efficient and secure communication across their IT landscape.

A.1.4.5 Spring Boot

Spring Boot is an open-source framework designed to simplify the development of stand-alone, production-grade Spring-based applications. One of its key features is auto-configuration, which automatically configures your application based on the dependencies you include. This significantly reduces the need for extensive manual configuration, allowing developers to focus more on writing business logic rather than boilerplate code. Additionally, Spring Boot includes embedded servers like Tomcat, Jetty, and Undertow, enabling you to run your applications without needing to deploy WAR files. This embedded server capability streamlines the development and deployment process, making it easier to get applications up and running quickly.

Another notable feature of Spring Boot is its starter POMs (Project Object Models). These starter POMs simplify dependency management by grouping commonly used dependencies into convenient packages. This approach not only saves time but also ensures that your project includes all necessary dependencies for a given functionality. Spring Boot also offers production-ready features such as metrics, health checks, and externalized configuration. These features help ensure that your application is ready for production environments, providing insights into application performance and health.

Spring Boot is particularly well-suited for developing microservices architectures. Its modular design and ease of integration with other Spring projects and third-party libraries make it an excellent choice for building scalable and maintainable applications. The framework supports rapid development, allowing developers to quickly create and deploy microservices with minimal configuration. Additionally, Spring Boot 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 popular choice for modern application development. Whether you're looking to build a simple web application or a complex microservices system, Spring Boot provides the tools and capabilities to help you achieve your goals efficiently.

A.1.4.6 Node.js

Node.js is an open-source, cross-platform runtime environment that allows you to run JavaScript code outside of a web browser. It uses the V8 JavaScript engine, the same engine used by Google Chrome, to execute JavaScript code on the server side. Node.js is designed with an event-driven architecture, which efficiently handles asynchronous operations, making it ideal for applications requiring real-time processing and high concurrency. Its non-blocking I/O model allows it to manage multiple requests simultaneously without waiting for any single operation to complete.

Node.js comes with npm (Node Package Manager), the largest ecosystem of open-source libraries and packages, facilitating easy integration and development. This architecture supports the development of scalable applications, 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 like chat apps and online gaming, and creating microservices architectures. Its performance, unified development process using

JavaScript for both client-side and server-side, and large community support make Node.js a popular choice in modern web development.

Popular frameworks like Express.js and Koa.js further enhance its capabilities, providing robust features for web and mobile applications. Node.js is widely used for its efficiency, scalability, and versatility, making it a valuable tool for developers looking to create high-performance applications.

Node.js provides several advantages when running on IBM Power Systems:

- Performance: Node.js's non-blocking I/O model ensures efficient handling of multiple requests, enhancing overall system performance.
- Scalability: The microservices architecture supported by Node.js allows applications to scale easily to meet growing demands.
- Cost-Effective: Modernizing with Node.js can be more cost-effective compared to complete rewrites or migrations.

Overall, Node.js provides a robust framework for modernizing IBM Power systems, enabling the development of high-performance, scalable, and efficient applications.

A.1.4.7 Single Sign On

Single sign-on, or SSO, is an authentication scheme that lets users log in once using a single set of credentials, and access multiple applications during the same session.

Single sign-on simplifies user authentication, improves the user experience and, when properly implemented, improves security. It’s used often to manage authentication and secure access to company intranets or extranets, student portals, public cloud service, and other environments where users need to move between different applications to get their work done. It’s also used increasingly in customer-facing web sites and apps–such as banking and e-commerce sites–to combine applications from third-party providers into seamless, uninterrupted user experiences.

How single sign-on works

Single sign-on is based on a digital trust relationship between service providers – applications, web sites, 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 into one of the service providers, or into a central portal (such as an company intranet or college student portal) using SSO login credentials.
2. When the user is successfully authenticated, the SSO solution generates a session authentication token containing specific information about the user's identity—a username, email address, etc. This token is stored with 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 if user is already authenticated for the session. If so, 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 not, the user is prompted to reenter login credentials.

SSO variations

The SSO process described above – a single log-in and set of user credentials providing session access to multiple related applications – is sometimes called simple SSO or pure SSO. Other types of SSO include:

- Adaptive SSO

  Adaptive SSO requires an initial set of login credentials, but prompts for additional authentication factors or a new login when additional risks emerge – such as when a user logs in from a new device or attempts to access particularly sensitive data or functionality.

- Federated identity management (FIM)

  Federated identity management, or FIM, is a superset of SSO. While SSO is based on a digital trust relationship among applications within a single organization's domain, 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 (e.g., Slack or WebEx) without an additional log-in, or with a simple username-only log-in.

- Social login

  Social login enable end users to authenticate with applications using the same credentials they use to authenticate with popular social media sites. For third-party application providers, social login can discourage undesirable behaviors (false logins, shopping cart abandonment) and provide valuable information for improving their apps.

Related technologies

SSO may be implemented using any of several authentication protocols and services.

- SAML/SAML 2.0

Security Assertion Markup Language, or 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/OAuth 2.0

  Open Authorization, or OAuth, is an open standard protocol that exchanges authorization data between applications without exposing the user's password. OAuth enables using a single log-in to streamline interactions between applications that would typically require separate logins to each. For example, OAuth makes it possible for LinkedIn to search your email contacts for potential new network members.

- OpenID Connect (OIDC)

  Another open standard protocol, OICD uses REST APIs and JSON authentication tokens to enable a web site or application to grant users access by authenticating them through another service provider.

  Layered on top of OAuth, OICD is used primarily to implement social logins to third-party applications, shopping carts, and more. A lighter-weight implementation, OAuth/OIDC is often to SAML for implementing SSO across software-as-a-service (SaaS) and cloud applications, mobile apps, and Internet of Things (IoT) devices.

- LDAP

  Lightweight directory access protocol (LDAP) defines a directory for storing and updating user credentials, and a process for authenticating users against the directory. Introduced in 1993, LDAP is still the authentication directory solution of choice for many organizations implementing SSO, because LDAP lets them provide granular control over access the directory.

- ADFS

  Active Directory Federation Services, or ADFS, runs on Microsoft Windows Server to enable federated identity management – including single sign-on – 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 trouble. For example: Instead of logging into multiple applications multiple times per day, with SSO corporate end users can log into the corporate intranet just once for all-day access to every application they need. But by reducing significantly the number of passwords users need to remember, and the number of user accounts administrators need to manage, SSO can provide a number of other benefits.

- Reduced password fatigue

  Users with lots of passwords to manage often lapse into the bad and risky habit of using the same short, weak passwords—or slight variations thereof—for every application. A hacker who cracks one of these passwords can easily gain access to multiple applications. SSO lets users consolidate multiple short weak passwords into one single, long, strong password that's easier for users to remember and much more difficult for hackers to break.

- Fewer password- and credential-related vulnerabilities

  According to the IBM X-Force® Threat Intelligence Index 2024, 2023 saw a 71% year-over year increase in cyberattacks that used stolen or compromised credentials. SSO can reduce or eliminate the need for password managers, passwords stored in spreadsheets, passwords written on sticky notes and other memory aids—all of which provide targets for hackers or make passwords easier for the wrong people to steal or stumble upon.

- Fewer help desk calls

  According to industry analyst Gartner, 20 to 50 percent of IT help desk calls are related to forgotten passwords or password resets. Most SSO solutions make it easy for users to reset passwords themselves, with help desk assistance.

- Simplified security management

  SSO gives administrators simpler, more centralized control over account provisioning and access permissions. When a user leaves the organization, administrators can remove permissions and decommission the user account in fewer steps.

- Improved regulatory compliance

  SSO can make it easier to meet regulatory requirements around protection of personal identity information (PII) and data access control, as well as specific requirements in some regulations—such as HIPAA—around session time-outs.

SSO security risks

The chief risk of SSO is that if a user's credentials are compromised, they can grant an attacker access to all or most of the applications and resources on the network. But requiring users to create long and complex passwords—and carefully encrypting and protecting those passwords wherever they're stored—goes a long way toward preventing this worst-case scenario.

In addition, most security experts recommend two-factor authentication (2FA) or multi-factor authentication (MFA) as part of any SSO implementation. 2FA or MFA require users to provide at least one authentication factor in addition to a password – passcode sent to a mobile phone, a fingerprint, an ID card. Because these additional credentials are ones that hackers can not easily steal or spoof, MFA can dramatically reduce risks related to compromised credentials in SSO.

A.2 IBM

This section highlights IBM solutions to help you modernize your infrastructure and applications.

A.2.1 System Management

Modernizing your infrastructure often requires new approaches to system management, including using AI enhanced observability, application resource management and cloud ready solutions to assist you in refactoring your applications.

A.2.1.1 IBM Power Private Cloud with Shared Utility Capacity

IBM Power Private Cloud with Shared Utility Capacity (also known as Power Enterprise Pools 2 or PEP2) offers enterprises 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 Systems, optimizing utilization 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.

One of the key aspects of PEP2 is its integration with the Cloud Management Console (CMC), which provides a centralized interface for managing resource pools, monitoring usage, and handling capacity credits. This setup ensures that enterprises can efficiently manage their IT infrastructure and respond to changing business needs.

Integration with Red Hat OpenShift allows for seamless deployment and management of containerized applications on IBM Power Systems, enhancing scalability and accelerating modernization efforts. This makes the platform ideal for organizations adopting hybrid cloud strategies and looking to modernize their IT environments without sacrificing performance or control.

A.2.1.2 Turbonomic

Turbonomic is the premier solution for Application Resource Management (ARM), a hierarchical, application-driven approach that continuously analyzes applications' resource needs and generates fully automatable actions to ensure applications always get what they need to perform. It runs 24/7/365 and scales with the largest, most complex environments.

To perform Application Resource Management, Turbonomic represents your environment holistically as a supply chain of resource buyers and sellers, all working together to meet application demand. By empowering buyers (such as VMs) with a budget to seek the resources that applications need to perform, and sellers (such as hosts) to price their available CPU, memory, storage, and other resources based on utilization in real time, Turbonomic keeps your applications in an optimal state.

Turbonomic is a microservices architected platform that runs in your network or in the public cloud. It discovers and monitors your application environment through targets. It then performs analysis, anticipates risks to performance or efficiency, and recommends actions to avoid problems before they occur.

Documentation for Turbonomic can be found at `https://www.ibm.com/docs/en/tarm`.

A.2.1.3 Instana

IBM Instana Observability (Instana) is an observability platform that helps you analyze and troubleshoot microservices and containerized applications. It provides automated application performance monitoring, end-user experience monitoring, root cause analysis, and anomaly detection. With Instana, you can gain complete visibility into the health and performance of your applications and services.

Instana automatically makes your applications and services visible, provides context to that observed information, and then empowers you to take intelligent action based on that information.

- Automates discovery and visibility: Instana automatically discovers and monitors your applications, services, infrastructure, web browsers, mobile applications, and more for over 200 domain-specific technologies. In addition, it displays real-time data through distributed tracing and 1-second metrics.
- Provides context: Instana automates dependency mapping across the full stack for flexible application perspectives and provides powerful and easy-to-use data analytics. It puts performance data in context to deliver rapid issue prevention and remediation. You can drill down to generate new insights with endless flexibility from the entire repository of application-request trace data.
- Enables intelligent decision-making: Instana informs you whenever your customers are impacted by performance or stability issues in your applications within a few seconds of impact. In addition, Instana automates root-cause analysis by using event correlation, performance thresholds, errors, changes, and analysis of service level agreement (SLA) violations.

Extensive documentation for Instana can be found at
https://www.ibm.com/docs/en/instana-observability/current.

A.2.2 Cloud Paks

IBM Cloud Paks are a set of integrated software solutions designed to help businesses accelerate their digital transformation journey. Built on top of Red Hat OpenShift, IBM Cloud Paks provide a robust, cloud-native platform that enables enterprises to modernize their applications, manage workloads, and optimize resources across hybrid cloud environments. Each Cloud Pak is a comprehensive, pre-integrated package of software, tools, and services that address specific business needs, such as data management, AI, security, and automation.

IBM Cloud Paks are available in several distinct offerings, each tailored to address particular use cases. The following Cloud Paks are discussed in this section.

- IBM Cloud Pak for Business Automation (CP4BA)
- IBM Cloud Pak for Data (CP4D)
- IBM Cloud Pak for Applications (CP4A)
- IBM Cloud Pak for Integration (CP4I)
- IBM Cloud Pak for Business Automation (CP4BA)
- IBM Cloud Pak for AIOps (CP4AioOps)

These Cloud Paks are built with open-source technologies and are designed for use in hybrid cloud environments, which means they can operate across both on-premises infrastructure and public clouds. IBM Cloud Paks offer key benefits such as simplifying IT management, enhancing scalability, reducing operational costs, and improving security and compliance. By leveraging Red Hat OpenShift as the underlying platform, Cloud Paks ensure a consistent and flexible environment for deploying and managing applications, regardless of where they are hosted.

A.2.2.1 CP4A

IBM Cloud Pak for Automation (CP4A) is an integrated suite of tools designed to help businesses automate a wide range of processes across their organization, from business operations and IT workflows to customer interactions and data management. CP4A combines several automation capabilities, including 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.

Some use cases for IBM Cloud Pak for Automation are:

- Customer Service Automation:
- Invoice and Document Processing
- Human Resources Automation
- Supply Chain Management
- Fraud Detection and Prevention

IBM Cloud Pak for Automation (CP4A) is a robust and comprehensive solution that helps businesses automate a wide variety of processes, enhancing operational efficiency, agility, and decision-making. With tools for robotic process automation (RPA), business process management (BPM), decision automation, content management, and AI-powered insights, CP4A empowers organizations to optimize their workflows, reduce costs, improve compliance, and deliver better customer experiences. Its cloud-native architecture and scalability ensure that businesses can meet the growing demands of the digital age while maintaining flexibility and adaptability across hybrid and multi-cloud environments.

A.2.2.2 CP4I

IBM Cloud Pak for Integration (CP4I) is an integrated suite of tools and services designed to help businesses streamline and simplify their integration processes across hybrid and multi-cloud environments. By offering a unified, flexible, and scalable platform, CP4I enables organizations to connect their applications, data, and services across different systems, whether on-premises, in the cloud, or in hybrid environments. This helps companies ensure seamless data flow, improve business agility, and accelerate the development of new digital capabilities.

Use Cases for IBM Cloud Pak for Integration

- Hybrid Cloud Integration
- Customer Experience Transformation
- Supply Chain Optimization
- IoT Integration
- Banking and Financial Services Integration

IBM Cloud Pak for Integration is a comprehensive, flexible, and scalable platform that helps businesses seamlessly integrate their 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 their digital transformation, improve operational agility, and create better customer experiences. With its built-in security features and scalability, CP4I is a future-proof solution that can evolve alongside the needs of modern businesses, helping them stay competitive in an increasingly digital and interconnected world.

A.2.2.3 CP4BA

IBM Cloud Pak for Business Automation (CP4BA) is an integrated suite of AI-powered tools and solutions designed to help businesses streamline and automate their operations, improve efficiency, and drive innovation. It combines a set of capabilities, including business process management (BPM), workflow automation, robotic process automation (RPA), decision automation, and content management, all within a unified platform. CP4BA helps organizations automate repetitive tasks, make data-driven decisions, and adapt to changing market conditions by simplifying complex business processes.

Some use cases for IBM Cloud Pak for Business Automation are:

- Customer Service Automation
- Supply Chain Optimization
- Claims Processing
- Human Resources Automation
- Compliance and Risk Management

IBM Cloud Pak for Business Automation is a powerful platform that enables businesses to drive digital transformation by automating their 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, all while scaling their automation efforts across hybrid cloud environments. Whether improving customer service, optimizing supply chains, or ensuring compliance, IBM Cloud Pak for Business Automation

A.2.2.4 CP4AIOps

IBM Cloud Pak for AIOps is an integrated, AI-powered platform designed to help businesses automate IT operations and enhance the performance, availability, and security of their systems. By leveraging artificial intelligence (AI) and machine learning (ML) technologies, Cloud Pak for AIOps provides organizations with proactive insights, intelligent automation, and predictive analytics, enabling IT teams to detect and resolve issues faster, optimize system performance, and prevent downtime.

Use Cases for IBM Cloud Pak for AIOps are:

- 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 into IT operations, enabling organizations to proactively monitor, manage, and optimize their IT infrastructure. By leveraging machine learning 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, Cloud Pak for AIOps helps organizations create a smarter, more resilient IT environment that can drive better business outcomes.

A.2.2.5 CP4D

IBM Cloud Pak for Data is an integrated data and AI platform that helps organizations accelerate their journey toward data-driven decision-making by bringing together data management, analytics, AI, and automation in a unified solution. Designed for hybrid and multi-cloud environments, it enables businesses to collect, organize, and analyze their data at scale while ensuring consistency, governance, and security across all their data assets. IBM Cloud Pak for Data simplifies the complexity of managing data pipelines, machine learning models, and analytics, making it easier for teams to derive insights and drive business value.

Some use cases for IBM Cloud Pak for Data are:

- Data Modernization
- AI and Predictive Analytics
- Data-Driven Decision Making
- Compliance and Risk Management

IBM Cloud Pak for Data provides a comprehensive, integrated platform that helps organizations manage and utilize their data more effectively. It combines powerful data management, AI, and analytics tools with robust governance and security features to enable data-driven decision making across the enterprise. By simplifying the complexities of working with data at scale, IBM Cloud Pak for Data accelerates AI adoption, improves business agility, and enhances operational efficiency in hybrid cloud environments. Whether used for modernizing legacy systems, developing AI models, or optimizing business processes, it provides the tools necessary to unlock the full potential of data.

A.2.3 Watsonx

watsonx is IBM's portfolio of AI products that accelerates the impact of generative AI in core workflows to drive productivity.

- Open your AI future

  Get the flexibility 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. And employ governance and security controls for easier compliance.

- Integrate to innovate

  Deploy AI with minimal disruptions to your systems or operations. Embed it into specific use cases to realize value quickly. And transform processes to increase productivity.

IBM watsonx solutions

IBM watsonx provides a wide range of portfolio to drive productivity from within.

- IBM watsonx Orchestrate™ - Say goodbye to busywork

  Increase productivity by easily creating, deploying and managing AI assistants and agents to automate and simplify business and customer-facing processes.

- IBM watsonx Code Assistant - Code smarter, not harder

  Accelerate your developers' productivity and reduce time to market by infusing AI into the entire application lifecycle to automate development tasks and streamline workflows.

- IBM watsonx.ai - Step into your AI studio

  Develop custom AI applications faster and easier with an integrated, collaborative, end-to-end developer studio that features an AI developer toolkit and full AI lifecycle management.

- IBM watsonx.data® - Trust your data, trust your decisions

  Manage, prepare and integrate trusted data from anywhere, in any format so you can unlock AI insights faster and improve the relevance and precision of your AI applications.

- IBM watsonx.governance® - Mitigate the risks. Meet the regulations.

  Automate governance to proactively manage AI risks, simplify regulatory compliance and create responsible, explainable AI workflows.

A.3 Open Source

Open source solutions on IBM Power Systems offer enterprises a powerful and flexible foundation for modern workloads, enabling the use of industry-leading technologies such as Linux, Kubernetes, and container orchestration platforms like Red Hat OpenShift. IBM Power supports popular open source databases, analytics tools, and development frameworks, optimized for the architecture's high performance and scalability. By combining the reliability and security of Power Systems with the innovation and cost-efficiency of open source software, organizations can build agile, cloud-ready environments that support digital transformation, DevOps practices, and hybrid cloud strategies.

A.3.1 KVM

KVM (Kernel-based Virtual Machine) is an open-source virtualization technology that enables Linux-based operating systems to run virtual machines (VMs) on hardware. KVM is part of the Linux kernel and turns it into a hypervisor, allowing the operating system to run multiple isolated virtual environments on a single physical machine. KVM allows organizations to virtualize their IT infrastructure, enabling the running of different guest operating systems (including Linux, Windows, and others) alongside the host operating system.

On POWER10, KVM is introduced as a virtualization option within a PowerVM logical partition (LPAR). This means that you first create a Linux LPAR using PowerVM, and then within that Linux LPAR, you can run KVM to host other Linux virtual machines (guests). KVM on IBM POWER10 combines the advantages of KVM (Kernel-based Virtual Machine) virtualization with the high-performance, scalability, and enterprise-grade features of IBM's 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 leverage cutting-edge hardware while benefiting from open-source, high-performance virtualization capabilities.

A.3.2 Artificial Intelligence Solutions

The ability to utilize AI to provide additional customer insights or to manage complex environments is critical to your modernization effort. This section describes some of the AI software solutions that are designed to utilize the AI acceleration technologies built into the IBM Power hardware.

A.3.2.1 RocketCE and Rocket AI Hub for IBM Power

RocketCE for IBM Power is a set of AI and data science packages built and supported for the IBM Power architecture. Packages include best-of-breed open-source AI tools that are Power10-optimized, leveraging Power10’s on-chip acceleration and are available via Rocket Software’s public Anaconda channel (https://anaconda.org/rocketce/repo).

Rocket AI Hub for IBM Power is an integrated and freely available set of best-of-breed open-source AI platform tools optimized for IBM Power such as:

- Katib
- Kubeflow
- Kubeflow Pipelines

– KServe
– RocketCE

As a platform approach, all tools are delivered as container images that are operated within Kubernetes-based environments such as Red Hat OpenShift. All tools are integrated via Kubeflow, are optimized to leverage unique AI hardware capabilities of the IBM Power platform, and optional commercial support is available.

A.3.2.2 Ollama

Ollama is an open-source project that lets you run large language models (LLM) on your own hardware. Ollama supports a list of models available on ollama.com/library including the IBM granite dense family. The IBM Granite 2B and 8B models are text-only dense LLMs trained on over 12 trillion tokens of data and are designed to support tool-based use cases and for retrieval augmented generation (RAG), streamlining code generation, translation and bug fixing.

Table A-1 describes some of the models that are supported with 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

Download binary from ollama.com and you can install the binary on your hardware.

To start the Ollama service run the command shown in Example A-1.

Example: A-1 Start Ollama service

```
# ollama serve
```

To pull and run model (get the models https://ollama) run the command shown in Example A-2.

Example: A-2 Run model

```
# ollama run granite3.1-dense.8b
>>>
```

Ollama creates an isolated environment to run LLMs locally on your system and includes all the necessary components for deploying AI models, such as model weights, configuration files, and necessary dependencies.

Here are some examples of how Ollama can help.

- Creating local chatbots

  With Ollama, developers can create highly responsive AI-driven chatbots that run entirely on local servers, ensuring that customer interactions remain private.

- Conducting local research

Universities and data scientists can leverage Ollama to conduct offline machine-learning research. This lets them experiment with datasets in privacy-sensitive environments, ensuring the work remains secure and is not exposed to external parties.

- Building privacy-focused AI applications

  Ollama provides an ideal solution for developing privacy-focused AI applications that are ideal for businesses handling sensitive information.

- Integrating AI into existing platforms

  Ollama can easily integrate with existing software platforms, enabling businesses to include AI capabilities without overhauling their current systems.

A.3.2.3 Streamlit

Streamlit is an open-source Python framework for data scientists and AI/ML engineers to deliver dynamic data apps with only a few lines of code. It provides a range of features for building data science applications, including support for data visualization, machine learning, and deep learning. Streamlit also provides a range of tools for deploying and sharing applications, making it easy to create and share data science projects with others.

Streamlit is also designed to be flexible and scalable. It can be used to build a wide range of applications, from simple data analysis tools to complex machine learning models. Additionally, Streamlit provides a range of tools for deploying and sharing applications, making it easy to create and share data science projects with others.

In terms of use cases, Streamlit is well-suited for a variety of applications, including data science projects, machine learning models, and web applications. Its ease of use and flexibility make it an ideal choice for both experienced and novice developers, making it a popular choice for building data science applications.

A.3.2.4 Llama.cpp

Llama.cpp, a C/C++ library designed for efficient inference of large language models (LLMs), can leverage the high performance and scalability of IBM Power systems. This integration allows for processing GGML-formatted models, such as Meta's LLaMA, Vicuna, or Wizard, without requiring a GPU.

To set up llama.cpp on IBM Power10, you need to ensure you have the necessary prerequisites, such as gcc-toolset-13. The library can be built from source using cmake, and it supports various BLAS backends like OpenBLAS for optimized performance. Once set up, llama.cpp can efficiently tokenize prompts and generate responses using top-K and top-P sampling algorithms. This setup enables enterprises to leverage IBM Power's high performance for AI workloads, enhancing the efficiency and scalability of their AI applications[1].

A.3.2.5 Trovares

Trovares, now known as Rocketgraph, offers the xGT graph analytics platform, which is highly compatible with IBM Power systems. Rocketgraph xGT is designed to handle very large and complex graph problems, allowing 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 significant computational performance advantages. IBM Power10 servers, with their superscalar, multithreaded, multi-core architecture and embedded AI acceleration technology, are ideal for high-performance workloads. Performance tests have shown that Rocketgraph xGT running

[1] https://community.ibm.com/community/user/blogs/amrita-h-s/2025/01/30/how-to-run-a-llm-model-on-ibm-power10-using-llamac

on IBM Power E1050 servers delivers faster results compared to x86 servers, making it an excellent choice for enhancing cybersecurity operations, fraud detection, and supply chain optimization.[2]

A.3.2.6 Equitus AI

Equitus AI is a US tech company specializing in AI-powered graph data analysis for government and commercial clients. They offer platforms like KGNN for unifying data into intelligent knowledge graphs and EVS for real-time video analytics. Their focus is on providing accurate, traceable, and secure AI solutions for complex data challenges.

KGNN runs natively on IBM Power10 servers empowering organizations to create autonomous AI systems that operate at the edge, independently of external cloud services, and overcoming GPU resource limitations. The on-premise Equitus KGNN appliance provides a robust solution for unifying and connecting knowledge assets residing within your organization's diverse systems and applications. This integration enables a holistic understanding of complex relationships and facilitates enhanced decision-making.

A.3.2.7 OpenTech

OpenXAI-Opentech is a Saudi Arabian tech company specializing in on-premise AI chatbot platforms (OpenXAI) and custom drones. They prioritize data security and use IBM Power for their AI infrastructure. They also develop AI for smart home applications. OpenTech leverages IBM Power infrastructure, particularly the Power10 systems, to deliver their AI-powered chatbot solutions with a focus on performance, security, and the ability to operate on-premise for data-sensitive applications.

A.3.2.8 ElinarAI

ElinarAI is an advanced AI solution designed to automate manual processes and enhance data management across various industries. It leverages cognitive AI to interpret and transfer information, reducing errors and improving efficiency. ElinarAI can handle tasks that require human cognition, such as processing large volumes of data, handling sensitive material, and performing complex analyses.

For investigators, ElinarAI offers specialized capabilities to automate the analytics of unstructured data. It enables the creation of Investigation Specific AIs (ISAIs) to scan and recognize entities on a large scale with high accuracy. This is particularly useful for law enforcement and intelligence communities dealing with extensive data sets, such as in cases of money laundering, tax evasion, and fraud.

ElinarAI has a strong relationship with IBM, with their solutions often augmenting IBM products like Cloud Pak for Automation and Watson Discovery. Their platform is also optimized to run on IBM Power infrastructure.

A.3.2.9 Wallaroo

Wallaroo AI is a platform designed to simplify and accelerate the deployment and management of machine learning models in production. It helps businesses operationalize AI by providing tools for:

- Deployment: Streamlining the process of getting models into production environments. This includes support for various model frameworks, hardware architectures, and cloud environments. Wallaroo automates model packaging and provides self-service deployment capabilities.

[2] https://community.ibm.com/community/user/blogs/jenna-murillo/2025/02/18/introducing-rocketgraph-xgt-on-ibm-power

- Inference: Efficiently running models to generate predictions. Wallaroo focuses on high-performance batch and real-time inference, with optimized resource utilization.
- Observability: Monitoring model performance and identifying issues. Wallaroo provides tools for tracking metrics, detecting drift, and generating reports and alerts.
- Scalability: Handling increasing volumes of data and traffic. Wallaroo is designed to scale AI deployments to meet the demands of real-world applications.

Wallaroo aims to address the challenges of putting AI into practice, enabling organizations to more effectively leverage machine learning for real-world applications. It emphasizes efficiency, flexibility, and ease of use, with features like automated workflows, support for diverse use cases, and integrated tools.

For more information refer to https://wallaroo.ai/.

A.3.2.10 Finacle AI

IBM and Finacle have 20+ years of relationship. IBM Power is a trusted infrastructure for Finacle. 70% of global deployments trust IBM Power with their Finacle core banking. As banks look at how to efficiently meet strategic objectives such as improving customer experience, driving personalization, improving operations and speed to market, they are looking to AI.

To support customers as they look to adopt AI, Finacle introduced the Finacle AI Platform which is an innovative solution designed with explainability, security, and ease of use features. Finacle AAI attaches to the core banking application to enable customers to seamlessly infuse AI into the banks' digital operations and derive actionable insights. The Finacle AI Platform empowers banks to accelerate their AI journeys by offering a banking specific AI platform that delivers:

- Low-code solution enables business users to leverage AI
- Pre-built predictive analytics use cases for different business lines, users, and customer situations
- Services to design, build, and deploy custom AI use cases

Deploying the Finacle AI platform on IBM Power greatly simplifies integration with core banking, lifecycle management, capacity management, security and compliance. Figure 10-7 shows a deployment architecture on IBM Power.

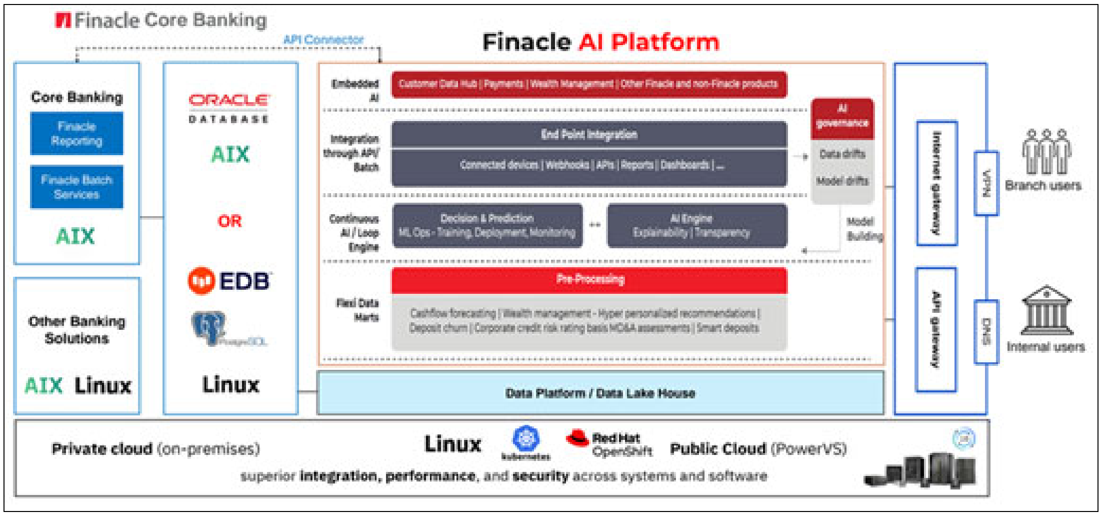

Finacle Core Banking
API Connector
Finacle AI Platform
Core Banking
Finacle Reporting
Finacle Batch Services
AIX
Other Banking Solutions
AIX Linux
ORACLE DATABASE
AIX
OR
EDB
PostgreSQL
Linux
Embedded AI
Customer Data Hub | Payments | Wealth Management | Other Finacle and non-Finacle products
Integration through API/ Batch
End Point Integration
Connected devices | Webhooks | APIs | Reports | Dashboards | ...
Continuous AI / Loop Engine
Decision & Prediction
ML Ops - Training, Deployment, Monitoring
AI Engine
Explainability | Transparency
Flexi Data Marts
Pre-Processing
Cashflow forecasting | Wealth management - Hyper personalized recommendations |
Deposit churn | Corporate credit risk rating basis MD&A assessments | Smart deposits
AI governance
Data drifts
Model drifts
Model Building
Data Platform / Data Lake House
Internet gateway
API gateway
VPN
DNS
Branch users
Internal users
Private cloud (on-premises)
Linux
kubernetes
Red Hat OpenShift
Public Cloud (PowerVS)
superior integration, performance, and security across systems and software

Figure 10-7 Sample deployment on IBM Power for Finacle AI Platform

A.3.3 Pipelines

Pipelines are automated workflows used in Continuous Integration and Continuous Deployment (CI/CD) to streamline and standardize the process of building, testing, and deploying applications. They automatically integrate every code change, run tests, and deliver updates to production environments, reducing manual effort and minimizing the risk of errors. A typical pipeline includes stages like source code integration, automated testing, artifact generation, and deployment, helping teams deliver software faster and more reliably. By implementing CI/CD pipelines, organizations can enhance efficiency, consistency, and agility throughout the development lifecycle. The following section explores CI/CD pipelines commonly used in modernizing IBM Power infrastructure.

A.3.3.1 GitHub

GitHub is a web-based platform that provides version control using Git, enabling developers to collaborate on projects more effectively. It offers a range of features such as repositories, branches, pull requests, and issues, which help manage and track changes to code. GitHub's collaborative tools allow multiple developers to work on the same project simultaneously, review each other's code, and merge changes seamlessly. Additionally, GitHub integrates with various CI/CD tools, making it easier to automate workflows and deploy applications.

Beyond version control, GitHub fosters a vibrant community of developers who contribute to open-source projects, share knowledge, and collaborate on innovative solutions. The platform supports various programming languages and frameworks, making it versatile for different types of projects. GitHub also offers GitHub Actions, a powerful automation tool that enables developers to create custom workflows directly within their repositories. With its extensive documentation and user-friendly interface, GitHub has become an essential tool for modern software development.

A.3.3.2 GitLab

GitLab is an all-in-one DevOps platform that streamlines the entire software development lifecycle by integrating tools for source code management, continuous integration and deployment (CI/CD), security, and monitoring into a single application. It enhances team collaboration through features like code review, issue tracking, and project management, while its powerful CI/CD capabilities automate testing and deployment to ensure rapid, reliable code delivery. GitLab also includes advanced security functions such as vulnerability scanning and compliance checks to help safeguard applications throughout their lifecycle.

As an open-source platform, GitLab benefits from a vibrant community of developers who contribute to its ongoing improvement. Its flexibility allows users to tailor the platform to their unique workflows and integrate with a wide range of external tools and services. With an intuitive user interface and extensive documentation, GitLab is a trusted choice for organizations aiming to accelerate DevOps adoption, boost productivity, and foster seamless collaboration across development teams.

GitHub and GitLab have very similar characteristics as they are both based on Git and provide many of the same features. However there are a several fundamental differences:

- GitLab is open source which means you can download the source code from here and self host the service on your own servers or on a cloud provider.
- GitLab offers its own deployment platform built on Kubernetes. With GitHub you would need to use an external platform, like AWS or Heroku and trigger your deploys there.
- GitLab has built-in Continuous Integration/Continuous Deployment (CI/CD) and DevOps workflows, making it a comprehensive solution for the entire software development

lifecycle. GitHub, on the other hand, requires integration with third-party CI/CD tools like Jenkins, CircleCI, or TravisCI.

A.3.3.3 Bitbucket

Bitbucket is a Git-based source code repository hosting service owned by Atlassian. It is another alternative to GitHub and GitLab. It provides a central platform for managing Git repositories, collaborating on code, and guiding development workflows. Bitbucket integrates seamlessly with other Atlassian tools like Jira and Trello, enhancing project management and team collaboration. It offers features such as pull requests, code reviews, and built-in CI/CD with Bitbucket Pipelines, allowing teams to automate testing and deployment. Bitbucket supports both cloud and self-managed data center hosting options, catering to different organizational needs.

Bitbucket Includes Bitbucket Pipelines for CI/CD, allowing teams to automate their build, test, and deployment processes. It features a user-friendly interface and strong collaboration tools, including pull requests and code reviews and is well-suited for teams using Atlassian's DevOps tools.

A.3.4 Databases

Modern databases are crucial for digital transformation, enabling businesses to efficiently store, manage, and analyze vast amounts of data. They power applications, facilitate real-time decision-making, and drive innovation through advanced features like scalability, flexibility, and diverse data support. By providing the foundation for data-driven insights, modern databases empower organizations to optimize operations, enhance customer experiences, and gain a competitive edge in the data-driven world.

A.3.4.1 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 various forms of data. As a NoSQL database solution, MongoDB does not require a relational database management system (RDBMS), so it provides an elastic data storage model that enables users to store and query multivariate data types with ease. This not only simplifies database management for developers but also creates a highly scalable environment for cross-platform applications and services.

MongoDB documents or collections of documents are the basic units of data. Formatted as Binary JSON (Java Script Object Notation), these documents can store various types of data and be distributed across multiple systems. Since MongoDB employs a dynamic schema design, users have unparalleled flexibility when creating data records, querying document collections through MongoDB aggregation and analyzing large amounts of information.

Over the years, MongoDB has become a trusted solution for many businesses that are looking for a powerful and highly scalable NoSQL database. But MongoDB is much more than just a traditional document-based database and it boasts a few great capabilities that make it stand out from other DBMS.

- Load balancing: MongoDB's load balancing sharing process distributes large data sets across multiple virtual machines at once while still maintaining acceptable read and write throughputs. This horizontal scaling is called sharding and it helps organizations avoid the cost of vertical scaling of hardware while still expanding the capacity of cloud-based deployments.
- Ad hoc database queries: MongoDB provides an ability to handle ad hoc queries that don't require predefined schema.

  - Multi-language support: MongoDB supports several popular programming languages, including Python, PHP, Ruby, Node.js, C++, Scala, JavaScript and many more.

MongoDB Editions

MongoDB comes in two main editions: Community Edition and Enterprise Edition.

- MongoDB Community Edition is free and open-source, suitable for learning and development. It includes the core database functionality.
- MongoDB Enterprise Edition is a commercial version that builds on the Community Edition by adding advanced security, management tools, and features for mission-critical deployments. These include:
  - Enhanced Security: Beyond basic authentication, Enterprise Edition offers LDAP and Kerberos integration, auditing, and encryption at rest (with KMIP) to meet stringent security requirements.
  - Advanced Management Tools: MongoDB Ops Manager automates tasks like deployment, upgrades, backups, and monitoring, streamlining database administration.
  - Additional Features: Enterprise Edition may also include features like the BI Connector for integrating with SQL-based business intelligence tools.
  - Support: Provides access to 24/7/365 technical support.

A.3.4.2 MariaDB

MariaDB is a community-developed, open-source relational database management system (RDBMS) that is compatible with MySQL. It is designed to be fast, reliable, and easy to use. MariaDB is built using the Linux kernel and provides a range of features for managing data, including support for multiple users and concurrent access, advanced indexing capabilities, and transactional support.

The key features of MariaDB are:

- High Performance

  MariaDB is known for its speed and efficiency, making it suitable for demanding workloads.
- Robustness

  MariaDB offers a stable and reliable platform with a proven track record.
- Active Community

  MariaDB benefits from a large and active community - providing support, resources and ongoing development.

MariaDB is well-suited for a variety of applications, including web applications, e-commerce sites, and data analytics platforms. Its speed and performance make it an ideal choice for handling large amounts of data, while its ease of use and flexibility make it a popular choice for both experienced and novice database administrators.

A.3.4.3 PostgreSQL

PostgreSQL is a powerful, open source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads. PostgreSQL comes with many features aimed to help developers build applications, administrators to protect data integrity and build fault-tolerant environments, and help you manage your data no matter how big or small the dataset.

PostgreSQL is an ideal database solution for enterprises in a variety of different industries.

- OLTP and analytics

PostgreSQL is great for managing OLTP (Online Transaction Processing) protocols. As a general purpose OLTP database, PostgreSQL works well for a variety of use cases like e-commerce, CRMs, and financial ledgers. PostgreSQL's SQL compliance and query optimizer also make it useful for general purpose analytics on your data.

- Geographic information systems

  PostGIS is an Open Geospatial Consortium (OGC) software offered as an extender to PostgreSQL. It allows PostgreSQL to support geospatial data types and functions to further enhance data analysis. By supporting geographic objects, PostgreSQL can refine sales and marketing efforts by augmenting situational awareness and intelligence behind stored data as well as help improve fraud detection and prevention.

- Database consolidation

  Move legacy databases to PostgreSQL while consolidating license costs, retiring servers, and cleaning up database sprawl. This can remove vendor-lock in, decrease the total cost of ownership for the databases, and improve application portability.

A.3.4.4 Fujitsu Enterprise Postgres

Fujitsu Enterprise Postgres is an enhanced, enterprise-grade PostgreSQL solution. It's designed for organizations that need strong query performance and high availability. Fujitsu builds on the open-source PostgreSQL system, adding features for improved security, performance, and management. Key enhancements include:

- Enhanced Security: Tools like Transparent Data Encryption and Data Masking.
- High Availability: Pre-configured clusters with automated failover.
- Performance Optimizations: Features like Vertical Clustered Index and Global Meta Cache.

Fujitsu Enterprise Postgres aims to provide a robust, cost-effective database solution with enterprise-level support.

A.3.4.5 Neo4J

Neo4j is a leading graph database management system designed to efficiently handle and query large, complex datasets in the form of graphs. Unlike traditional relational databases, which organize data in tables and rows, Neo4j stores data as nodes (representing entities) and relationships (representing connections between entities). This structure allows for more intuitive and flexible data modeling, making it ideal for applications where relationships are central, such as social networks, fraud detection, and recommendation systems. The query language used in Neo4j, called Cypher, is specifically optimized for graph operations, allowing users to express complex queries that traverse relationships with ease and in an intuitive manner.

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 execute queries that involve multiple hops or complex relationships, which would be slow and cumbersome in traditional relational databases. This makes Neo4j particularly suited for use cases such as network analysis, real-time recommendation engines, and knowledge graphs. Furthermore, Neo4j’s ACID-compliant transactional nature ensures 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 leveraging graph-based data.

A.3.4.6 EnterpriseDB

EnterpriseDB (EDB) is a leading provider of enterprise-class PostgreSQL solutions, offering robust database management tools and services. EDB's flagship product, EDB Postgres Advanced Server, enhances PostgreSQL with additional features such as performance optimization, security enhancements, and compatibility with Oracle databases. EDB also provides tools for high availability, disaster recovery, and monitoring, ensuring that organizations can maintain reliable and efficient database operations.

EDB's solutions are designed to support modern application development and deployment, including integration with cloud-native technologies like Kubernetes and Red Hat OpenShift. This allows organizations to leverage containerization and orchestration for scalable and resilient database environments. EDB's commitment to open-source innovation is evident in its contributions to the PostgreSQL community, driving advancements in features like incremental backups, JSON enhancements, and logical replication.

Running EDB on IBM Power Systems provides additional benefits, including enhanced performance, security, and scalability. IBM Power10 processor technology optimizes the execution of complex queries and supports large-scale data processing, making it ideal for enterprise workloads. The compatibility with Red Hat OpenShift on IBM Power Systems further enables organizations to deploy and manage containerized applications efficiently, leveraging the robust infrastructure of IBM Power for critical database operations.

For more information on running EDB on IBM Power see this link
https://community.ibm.com/community/user/viewdocument/ibm-power-solution-for-edb-postgres?CommunityKey=068c1cf5-2cf5-4d82-88c5-e067df2580bd

A.3.5 Frameworks

Frameworks are platforms that developers use to build software applications more efficiently. These frameworks provide reusable code, libraries, tools, and best practices that simplify common tasks like user interface design, database access, and communication between systems. This section describes some that are available on IBM Power.

A.3.5.1 .NET

.NET is a versatile and powerful framework developed by Microsoft for building a wide range of applications, including web, desktop, mobile, gaming, and IoT. It supports multiple programming languages such as C#, F#, and Visual Basic, providing developers with flexibility and choice. The framework includes a comprehensive class library and runtime environment, enabling efficient development and execution of applications. .NET Core, a cross-platform version of .NET, allows developers to create applications that run on Windows, macOS, and Linux, enhancing its reach and usability.

One of the key features of .NET is its support for modern development practices, including microservices architecture, cloud-native applications, and containerization. With integration into Azure, Microsoft's cloud platform, .NET enables seamless deployment and scaling of applications in the cloud. The framework also includes tools for automated testing, continuous integration, and continuous deployment (CI/CD), facilitating robust and efficient development workflows. Additionally, .NET's compatibility with popular development environments like Visual Studio and Visual Studio Code enhances productivity and collaboration among developers.

.NET's active community and extensive documentation provide valuable resources for developers, fostering innovation and knowledge sharing. The framework is continuously updated with new features and improvements, ensuring it remains relevant and capable of meeting evolving development needs. Microsoft's commitment to open-source development is evident in .NET's open-source nature, allowing developers to contribute to its growth and leverage its capabilities for diverse projects. Overall, .NET is a robust and adaptable framework that empowers developers to create high-quality applications across various platforms and industries.

.NET has increasing support and options for running on IBM Power Systems. This allows organizations that have invested in .NET technologies to leverage 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 (SLES), 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, allowing for containerized deployments on IBM Power.

A.3.5.2 Apache Kafka

Apache Kafka is a distributed streaming platform designed for building real-time data pipelines and streaming applications. At its core, Kafka operates as a publish-subscribe messaging system, allowing producers to publish streams of records to topics, and consumers to subscribe to these topics to receive the data. These topics are divided into partitions, which are ordered, immutable sequences of records. This partitioning enables horizontal scalability, allowing Kafka to handle massive volumes of data by distributing it across multiple brokers (servers) in a cluster. Each partition is typically replicated across multiple brokers to ensure fault tolerance and high availability.

The architecture of Kafka is built around a few key components. Brokers are the servers that host the partitions of topics. ZooKeeper is used to manage the Kafka cluster, tracking the status of brokers and partitions. Producers write data to topics, choosing which partition to write to (often based on a key). Consumers read data from topics, keeping track of their position (offset) within each partition. This decoupled architecture allows producers and consumers to operate independently and at different rates. Kafka's design prioritizes durability and reliability, ensuring that messages are persisted and delivered effectively, even in the face of broker failures.

The versatility of Apache Kafka has led to its widespread adoption across various industries. It is commonly used for building real-time data pipelines for analytics, data integration, and event-driven architectures. Use cases range from tracking user activity on websites and processing financial transactions to ingesting sensor data from IoT devices and powering real-time recommendation systems. Its scalability, fault tolerance, and ability to handle high-throughput streams of data make it a foundational technology for modern data infrastructure and real-time application development.

A.3.6 Security

Here are some Opensource security solutions available for IBM Power.

A.3.6.1 Keycloak

Keycloak is an open-source Identity and Access Management (IAM) tool developed by Red Hat. It provides authentication, authorization, and user management services for modern applications and services. Instead of building your own login and security logic, Keycloak offers a standardized, customizable, and secure solution out of the box.

Some key features of Keycloak are:

1. Single Sign-On (SSO)

   Users can log in once to access multiple applications without needing to authenticate again.

2. Identity Brokering and Social Login

   Keycloak supports identity federation with external providers (like Google, Facebook, or corporate identity systems), allowing users to log in using existing credentials.

3. Role-Based Access Control (RBAC)

   Define roles and permissions to control what users can access across applications and services.

4. User Federation

   Integrate with existing LDAP or Active Directory systems to manage users centrally.

5. Multi-Factor Authentication (MFA)

   Supports two-step authentication for enhanced security.

6. Admin Console and User Self-Service

   Admins can manage users, roles, and settings via a web UI, and users can manage their own passwords and profiles.

7. OAuth2, OpenID Connect, and SAML Support

   Keycloak supports all major authentication protocols, making it compatible with most modern applications and APIs.

A.3.6.2 Oauth

OAuth 2.0 is an authorization framework that decouples the role of the client from that of the resource owner. It relies on 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 hosting 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 requesting 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 presented by the client.

Different grant types define how the client obtains an access token (e.g., authorization code, implicit, 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 delegation of authorization, not authentication, and utilizes bearer tokens for accessing resources.

A.4 Independent Software Vendors

An Independent Software Vendor (ISV) is a company that creates, markets, and sells software applications designed to run on existing IBM platforms or operating systems. IBM works closely with ISVs to optimize their solutions for IBM's hybrid cloud and AI technologies. Through this collaboration, ISVs gain access to IBM's global network, technical resources, and go-to-market support, helping them accelerate innovation and grow revenue – while also increasing adoption of IBM's products and platforms.

A.4.1 Security

This section introduces some ISV solutions for security on IBM Power.

A.4.1.1 CyberVR

CyberVR is a cutting-edge cybersecurity platform that leverages virtual reality (VR) technology to enhance training and awareness. Developed by researchers from Sapienza University of Rome, CyberVR immerses users in interactive VR environments where they can engage with realistic cybersecurity scenarios. This approach aims to improve user understanding and retention of cybersecurity concepts by providing hands-on experience in a controlled, virtual setting. Studies have shown that CyberVR is not only as effective as traditional learning methods but also more engaging, leading to better outcomes in cybersecurity education.

Beyond its educational applications, CyberVR also serves as a tool for cybersecurity professionals to simulate and analyze potential threats in a virtual environment. By recreating realistic attack scenarios, users can develop and refine their response strategies without the risk of compromising actual systems. This dual-purpose functionality makes CyberVR a valuable resource for both learning and practical application in the field of cybersecurity.

CyberVR has recently expanded its capabilities to support IBM Power Systems, introducing the Isolated Recovery Environment (IRE) powered by its 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 impacting production systems. By creating high-fidelity, isolated replicas of entire IT environments, CyberVR allows teams to conduct live-fire 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 particularly valuable for industries with stringent operational resilience requirements, such as those outlined by the Digital Operational Resilience Act (DORA). With CyberVR's IRE, organizations can proactively identify and address vulnerabilities, ensuring robust protection for critical workloads and compliance with regulatory standards.?

A.4.1.2 Trend Vision One XDR security

Trend Vision One Security is a solution for IBM Power customers looking to protect 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, Trend Vision One aims to provide administration and DevOps teams greater control over their environment with central visibility and management. Utilizing Trend Vision One on IBM Power can help your organization modernize, simplify, and converge your security operations, enabling better protection against cyber threats across diverse hybrid IT environments.

Trend Vision One delivers real-time insights neatly displayed on your executive dashboard. No more manual tasks—just efficient, informed decision-making. While IBM Power frees up client resources, allowing them to focus on strategic business outcomes, Trend Vision One automates cyber security reporting and playbooks for more efficient and productive security operations. Security teams can stay ahead of compliance regulations, with real-time updates ensuring their enterprise security posture remains robust.

A solution brief can be found at
https://www.trendmicro.com/en_us/business/products/endpoint-security.html?modal=s7d-card-btn-ibm-power-sb-859b9e.

A.4.1.3 Precisely Enforcive (IBM i)

For end-to-end security and compliance management, Precisely's Enforcive Enterprise Security Suite is a comprehensive, easy-to-use security and compliance solution for IBM i. With over 20 fully integrated, GUI-controlled modules, the suite enables system administrators and security officers to manage security and compliance tasks efficiently and effectively – even managing multiple systems at a single time.

In today's world of privacy breaches, complex regulatory requirements and evolving threats, the Enforcive Enterprise Security Suite enables a comprehensive 'hardening' of your company's IBM i defenses against unauthorized access. Enforcive Enterprise Security Suite modules cover network security, authority swap, security monitoring, IBM i log transfer, and regulatory compliance. Additional modules such as Enforcive Field Encryption, Enforcive Password Self-Service, Enforcive Firewall Manager, among others can be added to tailor the solution to best meet the needs of your environment.

You can find more information on Enforcive at https://www.precisely.com/product/precisely-assure/enforcive-enterprise-security-suite.

A.4.2 Finance

A.4.2.1 IBM Financial Transaction Manager

IBM Financial Transaction Manager (FTM) on IBM Power offers a robust solution for financial institutions seeking to modernize their payment processing infrastructure. At its core, FTM is designed to integrate disparate payment systems, orchestrate complex transaction flows, and provide real-time monitoring and visibility across various payment types. These can include traditional methods like ACH, SEPA, and SWIFT, as well as newer, faster payment schemes. By acting as a central hub, FTM simplifies the complexities of managing multiple payment channels and diverse data formats, fostering a more streamlined and efficient operational environment.

Running FTM on IBM Power infrastructure brings significant advantages. IBM Power systems are renowned for their reliability, scalability, and robust security features, which are crucial for the high-stakes environment of financial transaction processing. The architecture of FTM is often modular and service-oriented, allowing institutions to adopt capabilities incrementally and integrate with existing legacy systems. Key features include support for industry standards like ISO 20022, real-time analytics and dashboards for operational insights, and the ability to handle high volumes of transactions with speed and precision. Furthermore, deployment on IBM Power often leverages virtualization and containerization technologies like Red Hat OpenShift, enhancing agility and flexibility in adapting to evolving business needs and regulatory landscapes.

More information on FTM can be found at https://www.ibm.com/products/financial-transaction-software.

A.4.2.2 Temenos

Temenos Group AG, a leader in the 2022 Gartner Magic Quadrant for Global Retail Core Banking. Temenos has partnered with IBM from past 20 years and with Red Hat from past 15 years to bring its core banking solution in to the market. There are thousands of customers running the traditional Temenos core banking on IBM Power. In second quarter of 2023, Temenos and IBM announced Temenos open platform to be available on Red Hat OpenShift running on IBM Power, for details see: https://www.temenos.com/news/2023/05/09/temenos-and-ibm-help-banks-accelerate-their-core-banking-modernization-with-hybrid-cloud/.

The solution enables clients to harness the advantages of hybrid cloud and speed up their digital transformation journey. By following Temenos' clear modernization path, banks can adopt a hybrid cloud strategy for their core banking systems and utilize emerging technologies such as Explainable AI and digital banking. Currently, Temenos Transact versions R23 and R24 are certified to run on Red Hat OpenShift with IBM Linux on the POWER platform.

A.4.2.3 In10s Technology

In10ns Technologies, also known as Intense Technologies, offers several banking solutions designed to enhance customer experience, streamline operations, and ensure regulatory compliance. Their flagship product, UniServe NXT, is an AI-driven platform that supports customer communication management (CCM), data management, and process automation. This platform helps banks deliver personalized communications, automate financial reconciliations, and improve data accuracy. UniServe NXT integrates cutting-edge cloud technologies, including Red Hat OpenShift and IBM Power infrastructure, to deliver outstanding performance and efficiency.

A.4.2.4 Fiserv

Fiserv is one of the world's leading financial services technology companies. The company's solutions empower more than 12,000 clients across more than 80 countries worldwide and help millions of consumers and businesses move and manage money quickly and conveniently. Fiserv offers a wide array of applications that are frequently deployed on IBM Power systems due to the platform's renowned reliability, security, and performance. These applications span various aspects of financial services, including core banking, payment processing, digital banking, risk and compliance, and data analytics.

In the realm of core banking, Fiserv's key platforms like Signature and Premier are often run on IBM Power. Signature, particularly its international version, is being modernized to leverage IBM's hybrid cloud capabilities, including Red Hat OpenShift on Power, to enhance agility and scalability. Premier also benefits from the robust and scalable nature of IBM Power, allowing financial institutions to efficiently manage core banking operations and integrate various add-on solutions.

A.4.3 GigaSpaces – Digital Integration Hub

GigaSpaces Technologies Inc. is a privately held software company founded in 2000 and headquartered in New York City, with additional offices in Europe, Asia, and Israel. The company specializes in providing high-performance computing solutions, focusing on real-time analytics, distributed computing, and middleware technologies. Over the years, GigaSpaces has developed several products to address the evolving needs of modern enterprises.?

One of their flagship offerings is the Smart Digital Integration Hub (DIH), a middleware solution designed to facilitate seamless software development and integration across diverse systems. Additionally, GigaSpaces has introduced GigaSpaces Cloud, a managed service on Google Cloud Platform, enabling organizations to leverage scalable and efficient cloud infrastructure. The company has also developed AnalyticsXtreme, a real-time analytics platform built on a data grid architecture, providing enterprises with the capability to process and analyze large volumes of data swiftly. These innovations position GigaSpaces as a key player in the realm of distributed computing and cloud-native technologies.

GigaSpaces has integrated its Digital Integration Hub (DIH) solutions with IBM Power Systems, enabling organizations to modernize their legacy IT infrastructure without the need for full system migrations. This approach allows enterprises to leverage existing IBM i

(AS/400), AIX, and Linux environments while adopting modern, cloud-native architectures. GigaSpaces' Smart DIH, when deployed on OpenShift clusters running on IBM Power, facilitates the creation of an event-driven, low-latency data fabric that decouples digital applications from core systems of record, thereby accelerating digital transformation initiatives. ?

Furthermore, GigaSpaces' enterprise retrieval-augmented generation (eRAG) platform, developed in collaboration with IBM's watsonx.ai and watsonx Assistant™, enhances the accessibility of structured enterprise data through natural language queries. This integration empowers users to interact with complex datasets in a more intuitive manner, improving decision-making processes. The combined capabilities of GigaSpaces' DIH and eRAG, supported by IBM's advanced AI tools, provide a robust framework for organizations seeking to innovate and modernize their operations on IBM Power Systems.

A.4.4 Pipelines

This section introduces some ISV developed pipeline solutions that run on IBM Power.

A.4.4.1 GitLab Runner & Red Hat OpenShift Operator

GitLab Runner is an application used to execute CI/CD jobs defined in your GitLab pipelines. It can run on various platforms and is responsible for fetching your code, running build/test scripts, and deploying applications.

On Red Hat OpenShift, GitLab Runner can be deployed as a containerized application, fully leveraging Kubernetes-native orchestration. This allows your CI/CD jobs to run efficiently within the same cluster where your applications are deployed, enabling faster builds and better resource utilization.

GitLab Runner is available as an operator for OpenShift specifically designed to simplify the deployment and management of GitLab Runners on OpenShift. Operators automate common operational tasks using custom resources.

Key benefits include:

- Automated lifecycle management: Install, upgrade, and scale GitLab Runners declaratively.
- Tighter GitLab integration: Easily register runners with GitLab using credentials stored in Kubernetes secrets.
- Custom resource definitions (CRDs): Manage runner configurations as code, just like other Kubernetes resources.
- Secure and isolated: Runs jobs within containers in OpenShift, taking advantage of security policies and namespaces.

A.4.4.2 Travis CI

Travis CI is a cloud-based continuous integration (CI) service that automates the process of building, testing, and deploying software projects. It is designed to work seamlessly with GitHub, Bitbucket, GitLab, and other version control platforms. Travis CI allows developers to define their 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, providing feedback on the success or failure of the code.

One of the key features of Travis CI is its support for parallel and multi-environment builds, enabling faster testing and deployment. It also integrates with various notification systems, such as email, Slack, and webhooks, to keep developers informed about the status of their builds. Travis CI's user-friendly interface and extensive documentation make it accessible for both beginners and experienced developers, helping teams streamline their CI/CD workflows and improve code quality.

Travis CI supports building and testing on multiple CPU architectures, including IBM Power Systems (ppc64le) and IBM Z (s390x). This capability allows developers to leverage Travis CI for continuous integration and deployment (CI/CD) workflows on IBM Power Systems, ensuring that applications are tested and deployed efficiently across different environments. Travis CI can operate both on-premises and in the cloud. Additionally, Travis CI's compatibility with Red Hat OpenShift on IBM Power Systems enables the deployment and management of containerized applications, leveraging the robust infrastructure of IBM Power for critical operations.

A.4.5 MuleSoft Anypoint Flex Gateway

To fully realize the value of enterprise data, businesses have often utilized application programming interfaces (APIs) to gain the benefits. APIs improve existing products, operations, and systems, open new streams of revenue, and provide richer insights that result in enhanced business strategies and provide richer customer experiences.

To transport data through APIs, however, requires a protection layer to ensure security of data and accessibility only to known actors. Mulesoft has partnered with IBM to provide Anypoint Flex Gateway on IBM Power.

A.4.5.1 Empowering integration on IBM Power

IBM Power is renowned for its robust performance, reliability, and scalability. With the native integration of Anypoint Flex Gateway, businesses using IBM Power servers can leverage one of the industry-leading API Management Platform from MuleSoft to seamlessly connect diverse systems, applications, and data sources. MuleSoft Anypoint Flex Gateway is an Envoy-based, ultra-fast, lightweight API gateway built on Envoy technology. Designed for seamless integration with DevOps and CI/CD workflows, Anypoint Flex Gateway delivers the performance needed for demanding applications and microservices, while ensuring enterprise-grade security and manageability across any environment.

A solution brief can be found at
https://www.mulesoft.com/sites/default/files/cmm_files/MuleSoft_AnypointFlexGateway_IBM%20Power_0.pdf

A.4.6 Backup & Recovery

A.4.6.1 Trilio Backup and Recovery

Trilio provides comprehensive backup and recovery solutions tailored for cloud-native environments, including Kubernetes and OpenStack. Designed to deliver tenant-driven backup, disaster recovery, and application mobility, Trilio ensures seamless protection and restoration of data across various platforms. Its solutions are built to minimize downtime and maximize productivity, offering features like intelligent recovery, granular file and folder restoration, and automated backup processes.

For Kubernetes, Trilio offers scalable, agentless backup and disaster recovery, enabling businesses to recover data in minutes and maintain compliance in highly regulated environments. The platform supports integration with Red Hat OpenShift and other Kubernetes distributions, facilitating seamless migration and recovery across clusters. Trilio's patented Continuous Restore feature significantly reduces recovery times, ensuring near-zero Recovery Time Objectives (RTO) and minimizing the financial impact of outages.

Trilio's solutions are also compatible with IBM Power Systems, leveraging the robust performance and scalability of IBM Power10 processor-based servers. This integration allows organizations to deploy and manage containerized applications efficiently, ensuring reliable and secure backup and recovery operations. By combining Trilio's advanced data protection capabilities with IBM Power Systems, enterprises can achieve enhanced resilience and business continuity for their critical workloads

A.4.6.2 Veeam Backup and Replication

Veeam Backup & Recovery is a comprehensive solution designed to protect data across diverse environments, including virtual servers, physical servers and workstations, cloud, and SaaS platforms. It operates by creating image-level backups of virtual machines, leveraging hypervisor snapshots for efficient data retrieval. For physical systems, it employs agents to capture the necessary data. These backups can be full copies or incremental, saving only the changed data blocks to optimize storage and speed up future backups. Veeam supports various backup methods like forward incremental, forever incremental, and reverse incremental, along with options for synthetic and active full backups to maintain backup chain health.

A cornerstone of Veeam's capabilities is its versatile recovery options, aimed at minimizing downtime. Instant VM Recovery allows users to boot a VM directly from a backup file within minutes, a critical feature for business continuity. For more granular recovery, Veeam offers file-level recovery, enabling the restoration of specific files and folders from guest operating systems. Application-aware recovery ensures consistency for critical applications like Microsoft Exchange, SQL Server, and Oracle, allowing for the recovery of individual application items. Veeam also supports VM replication, creating and maintaining ready-to-use copies of VMs for disaster recovery purposes, either on-site or off-site, with configurable failover points.

Beyond backup and recovery, Veeam incorporates features to enhance data management and resilience. Built-in data deduplication and compression reduce storage consumption and network bandwidth during transfers. Secure data transfer through encryption ensures the confidentiality of backup data both in transit and at rest. Veeam also provides robust monitoring and reporting tools, offering real-time insights into backup job status and infrastructure health. Its scalability allows it to adapt to environments ranging from small businesses to large enterprises. Furthermore, Veeam's support for immutable backups and integration with secure storage options like Veeam Vault strengthens defenses against ransomware and data loss, making it a key component of a comprehensive data protection strategy.

Veeam Backup & Replication and its associated components are fully supported on IBM Power systems, enabling enterprise-grade data protection for diverse workloads operating within this infrastructure. These integration capabilities are available for deployment in supported environments:

1. Veeam Agent for Linux on Power

Veeam offers a specific version of its Linux agent designed for IBM Power Systems. This agent supports Linux distributions running on IBM POWER9 and POWER10 architectures.

It utilizes a “nosnap” package, allowing operation without the Veeam kernel module by leveraging native file system snapshots.

This enables backup and recovery of files and directories on Linux-based workloads running on IBM Power.

2. Veeam Agent for IBM AIX

   Veeam provides a dedicated agent for the IBM AIX operating system, which is commonly used on IBM Power Systems.

   This agent allows for file-level backup of AIX Logical Partitions (LPARs) and can be installed in the root file system or within a specific Workload Partition (WPAR).

   It integrates with Veeam Backup & Replication, allowing centralized management and advanced recovery tasks.

   Supported AIX versions start from 6.1 Technology Level 5 (TL5).

3. Veeam Plug-ins for Enterprise Applications

   Veeam offers plug-ins for consistent backup and recovery of enterprise applications running on IBM Power, such as:

   - IBM Db2: Veeam provides a plug-in for IBM Db2 databases on Linux (including Linux on Power) and IBM AIX, ensuring 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.

4. Veeam Backup & Replication with IBM Cloud

   Veeam has strong integration with IBM Cloud, allowing IBM Power users to leverage the cloud for backup and disaster recovery.

   You can deploy Veeam Backup & Replication in IBM Cloud to protect on-premises Power workloads or protect workloads running natively in IBM Cloud. IBM and Veeam have a partnership to offer joint solutions for backup, migration, and disaster recovery on IBM Cloud, including support for hypervisor-level access for IBM Cloud VMware Solutions.

In summary, Veeam provides a comprehensive suite of solutions to protect various workloads running on IBM Power Systems, including Linux, AIX, and key enterprise applications, as well as offering integration with IBM Cloud and IBM Storage. This allows organizations using IBM Power to benefit from Veeam's robust backup and recovery capabilities.

A.4.6.3 Precisely Assure MIMIX

Precisely, is a software company specializing in data integrity tools. The company is known for its deep domain expertise and commitment to helping organizations make better decisions based on trusted data. Assure MIMIX is an automated high availability and disaster recovery 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 high availability solution for IBM AIX servers that can help eliminate downtime and enable data to be recovered from any point in time. It guards your business against unplanned and planned outages and lost data.

It works by paring efficient, real-time replication to a recovery server with continuous monitoring of cluster and application resources. If an outage is detected, applications are failed over to the recovery server – releasing storage resources, managing IP addresses, reestablishing replication, mounting file systems, and switching your application.

System availability is about more than keeping operations running after major disasters, planned or unplanned outages. “Micro” disasters, such as database corruption or data loss due to human error, can cost your business just as dearly. Which means affordable, manageable real-time disaster recovery protection for AIX applications and data is essential.

Assure MIMIX DR for AIX uses real-time replication and true continuous data protection (CDP) technology to deliver near-instantaneous data recovery from any point in time and enables production server rollback to recover from rolling disasters. It supports the recovery needs of any AIX environment, with efficient replication across any distance, between mixed storage and operating system versions, and between physical, virtual and cloud platforms. Assure MIMIX for AIX also integrates with IBM PowerHA for AIX to add point-in-time data recovery and offsite protection against regional or site outages to their PowerHA solution.

More information is available at
https://www.precisely.com/product/precisely-assure/assure-mimix-for-aix.

A.4.6.4 Assure MIMIX for IBM i

Assure MIMIX for IBM i is the leader in IBM i high availability (HA) and disaster recovery (DR). Thousands of companies worldwide, from small businesses to global enterprises, depend on Assure MIMIX to prevent data loss and eliminate planned and unplanned downtime.

Assure MIMIX provides full-featured, scalable real-time replication with extensive options for automating administration, comprehensive monitoring and alerting, customizable switch automation, and an easy graphical interface. Assure MIMIX works across any combination of IBM i server, storage, and OS versions. So, whether you need HA and DR protection for just one IBM i server, or your business depends upon a multi-site mix of on-premise, remotely hosted, and Cloud Service-based systems, Assure MIMIX delivers no-compromise data protection and business continuity that meets your needs today and the flexibility to support tomorrow’s business challenges.

Reliable, robust Disaster Recovery is essential for any business. True DR can only be achieved if the backup or replica copy of your data is maintained off-site, in a separate location that won’t be affected by any event that destroys the production data. So continuous, absolutely accurate replication of system and application data is essential for any business, of any size. Assure MIMIX leverages IBM’s Remote Journaling technology to ensure fast, efficient and unerringly accurate replication, between any combination of IBM i server or storage hardware, even when they are running different versions of iOS. It also takes the hard work and complexity out of setting up and managing replication using exclusive, flexible and highly automated Journal Centric Data Group technology.

Journal Centric replication intelligently monitors and responds as journaling for system objects and data is started or stopped, and even configures and starts replication for newly created journals. This is especially helpful in environments where journals are frequently created automatically by applications, or in larger, enterprise-scale operations where hundreds or even thousands of IBM i journals must be managed. Assure MIMIX always includes quick, easy installation and guided configuration; fast and efficient multi-threaded replication; convenient browser-based and 5250 management; mobile device-friendly graphical monitoring views; and many other must-have features and capabilities for complete, reliable no-compromise DR replication for every business.

More information is available at
https://www.precisely.com/product/precisely-assure/assure-mimix

A.4.7 Observability

These observability solutions provided by our ISVs are available for IBM Power.

A.4.8 Dynatrace OneAgent and Operator

Dynatrace is a monitoring platform that provides analytics and automation for unified observability and security. Dynatrace is built for uses cases such as Infrastructure observability, Application observability, Digital experience, Log analytics, Application security, Threat observability, Software delivery and Business analytics.

Dynatrace OneAgent discovers whatever processes running on the host. Based on what it finds, OneAgent automatically activates instrumentation specifically for the stack on the system.

A Dynatrace ActiveGate acts as a secure proxy between Dynatrace OneAgents and Dynatrace Clusters or between Dynatrace OneAgents and other ActiveGates – those closer to the Dynatrace Cluster. In addition to routing monitoring data captured by OneAgents, Dynatrace ActiveGate is also capable of performing monitoring tasks-using API 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 supports rollout and lifecycle of various Dynatrace components in Kubernetes and OpenShift.

As of launch, the Dynatrace Operator can be used to deploy a containerized ActiveGate for Kubernetes API monitoring. New capabilities will be added to the Dynatrace Operator over time including metric routing, and API monitoring.

An overview of the Dynatace monitoring stack can be found at
https://www.dynatrace.com/news/blog/dynatrace-observability-is-now-available-for-red-hat-openshift-on-the-ibm-power-architecture/

The certified Red Hat OpenShift operator is available at
https://catalog.redhat.com/software/containers/dynatrace/dynatrace-operator/60195e1e2937381f8e95740b?architecture=ppc64le

A.4.9 Splunk OpenTelemetry

Splunk OpenTelemetry (OTel) is an open-source observability framework designed to standardize 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 OpenTelemetry, 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 allows for 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 better support response from Splunk.

IBM has been working with Splunk to support the OpenTelemetry Collector on IBM Power Systems. This collaboration ensures that the collector can operate efficiently on IBM Power architecture (ppc64le).

A.4.10 Crest Infosolutions Alfresco

The Alfresco Digital Business Platform is a comprehensive, open-source platform designed to help organizations manage, govern, and utilize their content and processes effectively. It provides a suite of integrated services that enable businesses to build content-centric applications and streamline their operations. The platform is known for its modular architecture, scalability, and flexibility, allowing it to adapt to various industry needs and deployment scenarios, including on-premises, cloud, and hybrid environments.

At its core, the Alfresco Digital Business Platform encompasses 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, powered by Activiti, offers a Business Process Management (BPM) solution for automating workflows, managing tasks, and improving decision-making. Additionally, Alfresco Governance Services helps organizations manage their information lifecycle, ensure compliance, and handle records management according to regulatory requirements. The platform aims to empower businesses to unlock the value of their content, automate workflows, and ultimately drive digital transformation initiatives.

Crest Infosolutions is a global IT solutions and services provider with a strong focus on open-source technologies, and they are a prominent partner for Alfresco Digital Business Platform. They offer comprehensive services around Alfresco. Their expertise covers the full spectrum of Alfresco capabilities, such as document management, workflow and BPM, and records management, helping organizations to meet their digital transformation requirements.

Crest Infosolutions created a whitepaper which showed a significant performance advantage when running Alfresco Content Services on IBM Power versus an x86 solution.

B

Modernization using cloud native tools

This appendix describes how organizations can modernize IBM Power Systems workloads using cloud-native tools on Skytap on Azure. It discusses key Azure-native services that enable integration, improve scalability, and enhance automation for IBM i, AIX, and Linux for POWER environments.

In this appendix:

© Copyright IBM Corp. 2025.

B.1 Overview

Modernizing IBM i and AIX workloads requires a balanced approach that preserves the stability of legacy systems while adopting cloud-native technologies. Skytap on Azure enables organizations to migrate and run IBM Power Systems workloads, providing access to Azure-native tools for automation, DevOps, and AI-driven analytics.

Enterprise IT landscapes include a mix of IBM i, AIX, and x86 workloads. Traditionally, IBM i applications have operated in isolated environments, limiting agility and integration with modern development frameworks. Skytap on Azure bridges this gap, offering capabilities such as Live Clone for rapid environment duplication, integration with Azure DevOps for CI/CD, and compatibility with Azure-native networking, storage, and AI services.

By using Azure OpenAI Services and Semantic Vector Search, enterprises can extract value from structured and unstructured IBM i data, making it accessible for GenAI-driven business intelligence and automation. Additionally, ARCAD's integration with Azure DevOps allows IBM i application development, enabling teams to implement version control, automated testing, and continuous deployment while maintaining compliance and security standards.

This section explores how Skytap on Azure, combined with Azure-native tools, provides a scalable, cloud-integrated modernization approach for IBM Power Systems workloads.

B.2 Skytap on Azure

The following sections provide a comprehensive overview of Skytap on Azure, including its core capabilities, architectural design, security features, and real-world use cases.

B.2.1 What is Skytap

Skytap on Azure is a cloud-based Infrastructure as a service (IaaS) platform that enables enterprises to migrate and run IBM Power Systems workloads (IBM i, AIX, and Linux on Power) on Microsoft Azure. It provides a bare-metal IBM Power Systems compute environment, preserving the operational consistency of on-premises Power Systems environments while integrating Azure-native services for modernization. In May 2024, Kyndryl, the IT infrastructure services provider, acquired Skytap to enhance 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 Skytap regions. If your goal is to use Azure native tools, selecting Skytap on Azure is recommended.

B.2.1.1 Architecture overview

Skytap on Azure is engineered to facilitate the migration and operation of traditional IBM Power Systems and x86 workloads within the Microsoft Azure cloud environment. This integration enables organizations to use Azure's extensive cloud services while maintaining the performance and reliability of their existing applications. Below key architectural components are described:

IBM Power Systems integration

Skytap on Azure provides native support for IBM Power workloads, including IBM POWER9, and IBM POWER10, ensuring compatibility with enterprise applications in IBM i, AIX, and Linux on Power environments.

This is achieved through dedicated IBM Power hardware hosted in Azure data centers, preserving the performance characteristics required for mission-critical applications.

Software-Defined data centers (SDDC)

The platform utilizes SDDC technology to replicate on-premises data center environments. This includes the virtualization of infrastructure, storage, networking, operating systems, middleware, and applications, providing a comprehensive and flexible cloud environment.

Dynamic environment provisioning

Skytap enables the creation and management of dynamic environments, allowing users to easily clone, deploy, and manage multiple instances of their applications and systems.

This is particularly beneficial for development, testing, and training scenarios, where rapid provisioning and tear down of environments are required.

Networking and connectivity

The architecture supports various networking configurations to ensure secure and efficient connectivity:

- Azure Express Route Integration that provides private, high-bandwidth connections between on-premises environments and Azure, reducing latency and enhancing security.
- Virtual Network (VNet) peering that allows communication between Skytap environments and Azure services, facilitating hybrid application architectures.

Storage architecture

Skytap on Azure employs a scalable storage architecture designed for high availability and performance:

- ZFS-Based Storage Nodes utilize the Zettabyte File System (ZFS) for block-level storage, offering features such as data compression, snapshots, and cloning.
- Caching mechanisms to multiple layers of caching, including client OS, Virtual I/O Server (VIOS), and storage node caching, are implemented to optimize I/O performance

Security and compliance

The platform incorporates robust security measures to protect data and applications:

- Multi layered security controls are implementing security at physical, network, and application layers to safeguard against threats.
- Compliance certifications are in Skytap on Azure that meets industry compliance standards, including SOC 1, SOC 2, SOC 3, PCI DSS, ISO 27001, and GDPR compliance, ensuring security and regulatory adherence.

Figure B-1 on page 390 illustrates the architecture and connectivity model for Skytap on Azure. This hybrid design enables integration between on-premises data centers, Azure virtual networks (VNets), and Skytap environments running IBM Power and x86 workloads.

Organizations can securely connect their Skytap workloads to on-premises networks using Azure ExpressRoute, providing low-latency, high-bandwidth connectivity. Additionally, 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.

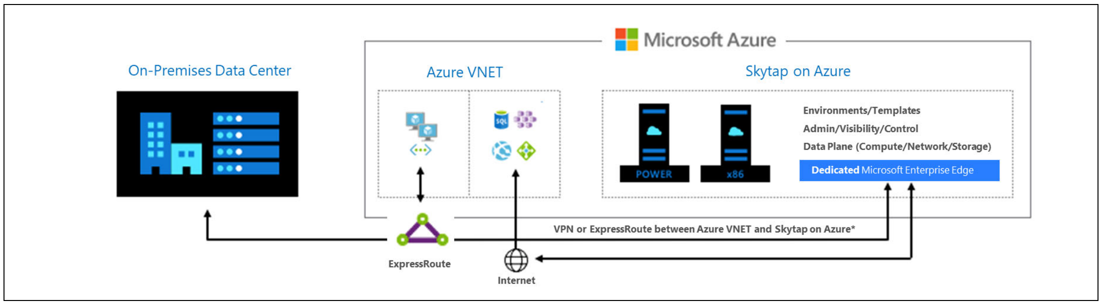


Figure B-1 Skytap on Azure Architecture

B.2.1.2 Use cases

Skytap on Azure helps businesses transition legacy workloads to the cloud, modernize applications, and enhance resilience. Key use cases include:

Datacenter exit

Migrate on-premises environments to Skytap with minimal disruption. Scale resources on demand, cut capital expenses (CapEx), and shift to 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 & application modernization with high availability

Automate provisioning, speed up innovation, and integrate cloud-native services for enhanced scalability and reliability.

Disaster recovery

Protect business operations by replicating on-premises environments in Skytap, ensuring rapid recovery and minimal downtime.

B.2.1.3 Case studies

Organizations across various industries have successfully used Skytap to modernize their IT infrastructures, migrate critical workloads, and enhance disaster recovery capabilities. To see further details, refer to the following win stories.

B.3 Azure native tools for modernization

Organizations modernizing IBM Power Systems workloads on Skytap benefit from native Azure services that enhance connectivity, storage, automation, and DevOps capabilities. These services enable integration between Skytap environments, on-premises infrastructure, and Azure-native resources, ensuring operational efficiency and scalability.

This section provides an overview of key Azure networking, storage, DevOps, automation, and AI-driven solutions that support workload modernization. It outlines best practices for integrating Azure ExpressRoute, VPN, and VNet peering for secure and optimized connectivity, Azure Blob Storage and Data Box Gateway for data management, and Azure DevOps with ARCAD for IBM i development workflows. Additionally, it explores automation with Terraform and Ansible, and the use of Azure AI services to improve operational intelligence and workload performance.

Using these Azure-native tools, organizations can optimize IBM Power workloads on Skytap, improve reliability, reduce complexity, and accelerate their cloud transformation journey.

B.3.1 Networking

Skytap on Azure supports various networking options to 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 displays the networking solutions in Skytap on Azure.

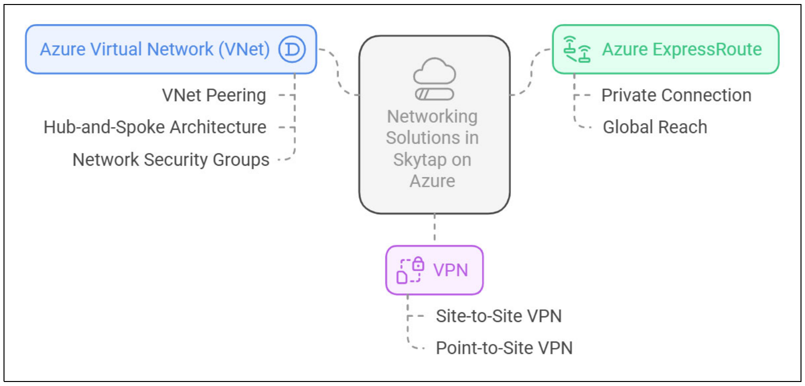


Figure B-2 Networking solutions in Skytap on Azure

B.3.1.1 Azure Express Route

Azure ExpressRoute provides a private, dedicated connection between on-premises environments, Skytap, and Azure, ensuring low-latency, high-bandwidth connectivity for mission-critical workloads.

Azure ExpressRoute Global Reach

Enables private connectivity across multiple ExpressRoute circuits, facilitating secure data exchange between on-premises, Skytap, and Azure regions. This enhances hybrid cloud architectures by allowing workloads to span across environments while maintaining operational consistency.

> **Note:** When using ExpressRoute from Skytap to an Azure-native region, there are no ingress/egress costs 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. Detailed pricing information is available `bandwitdh pricing`

B.3.1.2 VPN

Skytap supports Virtual Private Network (VPN) connectivity as a secure and cost-effective option for integrating on-premises environments with Azure. VPN tunnels encrypt data in transit, ensuring secure communication between Skytap environments and Azure-native services.

Site-to-Site VPN

Establishes an IPSec-based encrypted tunnel between on-premises infrastructure and Skytap in Azure, allowing 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

Provides managed VPN services for integrating Skytap environments with Azure Virtual Networks (VNets) while supporting hybrid cloud deployments.

B.3.1.3 VNet

Azure Virtual Network (VNet) is the fundamental building block for private networking in Azure. It enables 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 the need for a 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 (NSGs)

Apply security policies to control inbound and outbound traffic between Skytap, Azure services, and external networks.

B.3.2 Storage

Skytap on Azure provides integration with Azure native storage services for backup, archival, and disaster recovery. You can use these services to optimize data retention, ensure compliance, and facilitate workload recovery.

B.3.2.1 Azure blob storage

Azure Blob Storage is an object storage solution designed for scalable, secure, and cost-effective data retention. It supports multiple storage tiers, allowing 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 but still requires quick retrieval.
- Archive tier: Designed for long-term retention, requiring 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 disaster recovery.
- Offloading data for regulatory compliance and long-term archival.

B.3.2.2 AzCopy for data migration

AzCopy is a command-line utility for transferring large datasets to and from Azure Blob Storage efficiently. It supports multi-threaded uploads, resumption of interrupted transfers, and integration 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.

B.3.2.3 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:

- 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.

B.3.2.4 Backup and archival strategies

Organizations using Skytap on Azure can implement various backup and retention strategies to ensure data security and cost efficiency.

- Incremental backups store only the changed data since the last backup, reducing storage consumption and optimizing costs.
- Geo-Redundant Storage (GRS) replicates backup data to a secondary Azure region, providing resilience against regional failures and ensuring business continuity.

Policy-based retention automates data lifecycle management by transitioning backups between hot, cool, and archive storage tiers based on access frequency, reducing long-term storage expenses while maintaining compliance.

Figure B-3 displays the Skytap on Azure storage integration

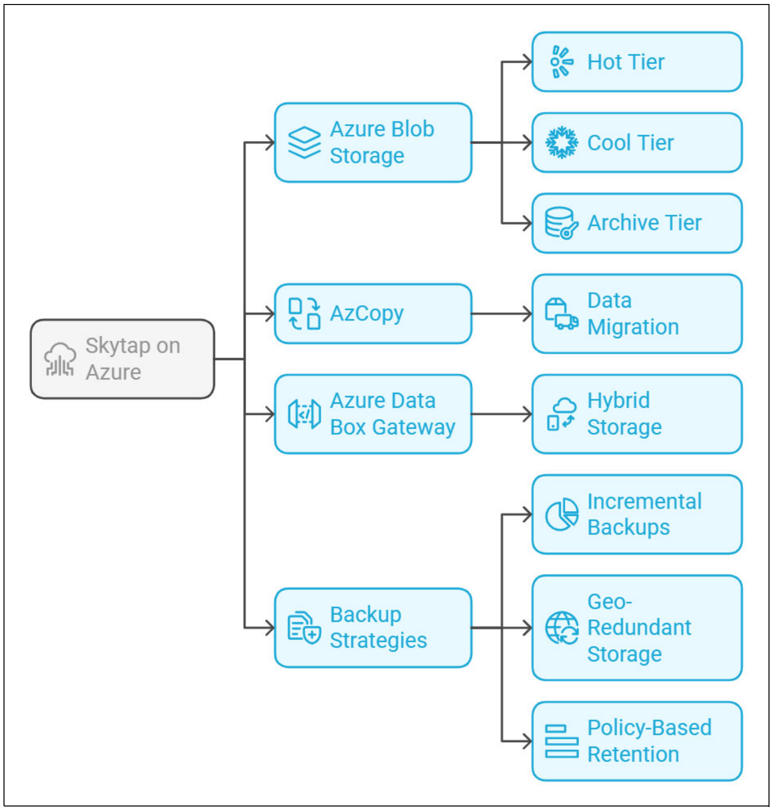


Figure B-3 Skytap on Azure storage integration

B.3.3 DevOps for IBM i

IBM i environments have traditionally used stable but isolated development practices with manual source control, development, and deployment. As enterprises move to DevOps, integrating IBM i into Git-based workflows, automated pipelines, and CI/CD processes is key for agility and efficiency.

Microsoft Azure DevOps offers tools for source control, CI/CD, and project management, helping IBM i teams modernize. However, integrating IBM i into a cloud-based DevOps framework involves cultural and technical challenges. Unlike open systems that rebuild and deploy full applications (which build and deploy full applications in .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 needs a unified toolset. Microsoft Azure DevOps offers a platform for IBM i and .NET developers to share repositories, pipelines, and deployment strategies collaboratively.
- Using Git-based source control with Azure Repos lets IBM i developers track versions, collaborate on code, and integrate with DevOps workflows. They can still use Rational Developer for i (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, cutting manual tasks and speeding up consistent releases.
- Managing IBM i workflows dynamically presents branching strategies for project-based development. Unlike static branching in open systems, dynamic branching allows teams to isolate features, apply fixes, and manage parallel development efficiently.

B.3.3.1 Native cloud components

Using IBM i development with Azure DevOps improves our software development process. This combined platform helps with planning, teamwork, and deployment, making tasks easier, enhancing code quality, and speeding up delivery. Figure B-4 shows the integration of IBM i with Azure DevOps

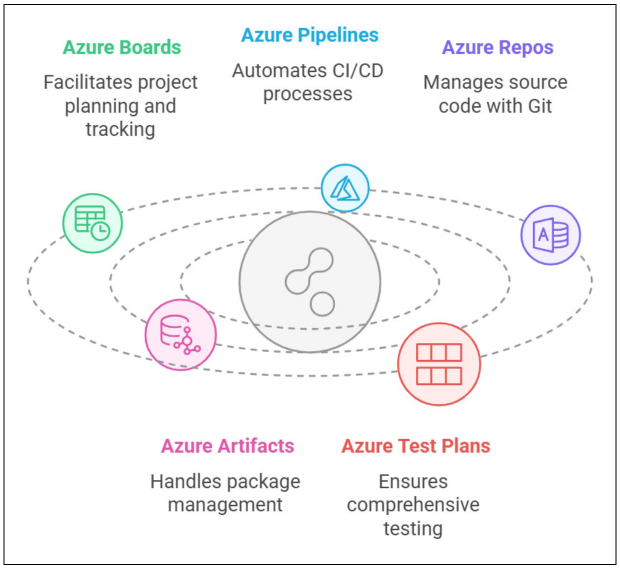

Azure Boards
Facilitates project planning and tracking
Azure Pipelines
Automates CI/CD processes
Azure Repos
Manages source code with Git
Azure Artifacts
Handles package management
Azure Test Plans
Ensures comprehensive testing

Figure B-4 Integrating IBM i with Azure DevOps

Azure Board

We utilize Azure Boards to plan, track, and discuss work across our teams. Its support for Kanban boards, backlogs, team dashboards, and custom reporting enables us to manage projects efficiently. This visual approach enhances collaboration among developers, testers, and project managers, ensuring alignment throughout the development process.

Azure Pipelines

Azure Pipelines facilitates continuous integration and continuous deployment (CI/CD) for our IBM i applications. By automating builds and deployments, we reduce manual errors, ensure consistent releases, and accelerate time-to-market. The pipeline's flexibility allows integration with various tools and services, supporting our diverse development and deployment needs.

Azure Repos

With Azure Repos, we manage our source code using Git repositories. This system provides version control, branching strategies, and pull requests, enabling collaborative development and code reviews. The teams can work concurrently on features and fixes, maintaining code integrity and facilitating seamless integration of changes.

Azure Artifacts

Azure Artifacts allows us to create, host, and share packages across our development teams. By integrating package management into our CI/CD pipelines, we simplify complex builds and ensure that our applications have consistent access to required dependencies. This approach enhances build reliability and promotes code reuse.

Azure Test plans

We employ Azure Test Plans to implement a comprehensive testing strategy, including manual, exploratory, and automated tests. This ensures our IBM i applications meet quality standards before deployment. Integrated test reporting and analytics provide insights into test coverage and application quality, guiding our continuous improvement efforts.

Note: A portion of IBM i customers have dependencies on x86 workloads, which include Windows and Linux environments[a]. Since Microsoft Azure is a cloud platform for Windows and Linux, many organizations using IBM i are likely to have an existing Azure subscription. This provides a logical entry point for use Azure DevOps tools for IBM i modernization, integrating with existing development pipelines, and bridging IBM i with cloud-native workflows.

[a] 2025 IBM i Marketplace Survey Results

B.3.3.2 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 continuous integration and continuous deployment (CI/CD) workflows.

Azure DevOps offers a unified toolchain for managing development, testing, and deployment processes. However, native DevOps tools lack IBM i-specific functionality, such as dependency builds, impact analysis, and RPG-specific version control. ARCAD bridges this gap by providing IBM i developers with the ability to work in Visual Studio Code (VS Code), Rational Developer for i (RDi), and even 5250 green screens, while leveraging Git-based repositories, automated builds, and deployment pipelines in Azure DevOps.

This section outlines how ARCAD integrates with Azure DevOps and how it modernizes IBM i development using cloud-native tools.

B.3.3.3 ARCAD architecture and tools

IBM i applications often operate within traditional environments, requiring structured modernization efforts. ARCAD integrates with Azure DevOps to automation, source control, testing, and deployment workflows while maintaining platform integrity.

The following tools enable structured modernization, continuous integration/continuous deployment (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, 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 ensures that code quality checks are embedded within CI/CD workflows.

ARCAD Builder (Automated IBM i Builds)

Application builds in open systems, IBM i follows a delta-based build approach, where only modified objects are compiled. 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, ensuring 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 is essential to ensure that code updates do not introduce unintended issues. ARCAD Verifier captures baseline program behavior and automatically revalidates functionality 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 (IBM 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, ensuring 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.

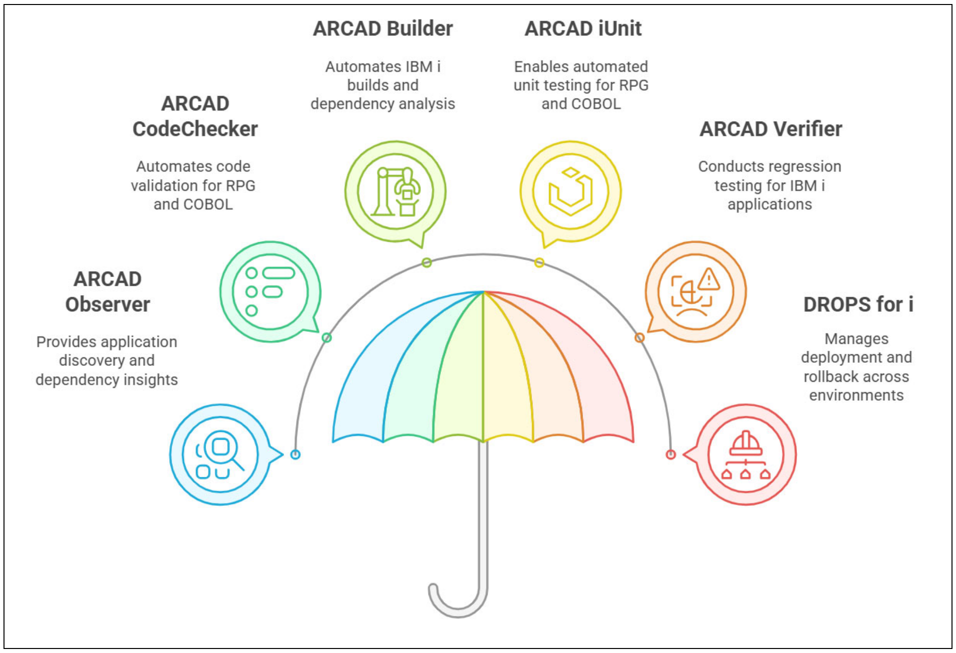

ARCAD Builder
Automates IBM i builds and dependency analysis
ARCAD iUnit
Enables automated unit testing for RPG and COBOL
ARCAD CodeChecker
Automates code validation for RPG and COBOL
ARCAD Verifier
Conducts regression testing for IBM i applications
ARCAD Observer
Provides application discovery and dependency insights
DROPS for i
Manages deployment and rollback across environments

Figure B-5 ARCAD tools for IBM i modernization

Key components for Azure DevOps with ARCAD

The following components helps IBM i modernization while maintaining compatibility with existing workloads:

- **Azure Repos** enable Git-based source control for IBM i applications, ensuring 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, RDi, or 5250 green screens for source code modifications.
  - Uses branch-based development, enabling teams to work in parallel on multiple features or bug fixes.
- **Azure Pipelines** automate CI/CD for IBM i applications, reducing manual intervention and improving deployment efficiency. ARCAD integrates with Azure Pipelines to:
  - Automate IBM i builds, ensuring 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** centralize 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** provide 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 UAT (User Acceptance Testing) execution using Azure DevOps workflows.

B.3.3.4 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 V25, 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 following capabilities shows 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, allowing developers to retrieve dependencies and application details efficiently
- Generates structured views of IBM i applications 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 allow metadata queries. Developers and analysts can search for dependencies, logic flows, and system documentation 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 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 utilizes different 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

ARCAD Discover AI assistant operates within secure environments, ensuring that IBM i applications are analyzed without exposing source code externally.Below shows the key considerations:

- AI processing remains within enterprise-controlled environments to maintain security
- No requirement to access external AI models, ensuring data privacy
- Deployment options include on-premises or private cloud instances based on compliance requirements.

> **Note:** The AI-driven insights generated by ARCAD Discover are processed locally or in controlled cloud environments, aligning with enterprise security policies.

Business impact and use cases

These 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 using AI-based scanning techniques
  - Ensures that AI processing remains within enterprise-controlled environments for compliance

B.3.4 Monitoring and analytics

Skytap on Azure integrates with Azure-native monitoring tools to centralize system logs, enable real-time analytics, and enhance operational visibility. These tools facilitate proactive management of IBM Power Systems workloads running on Skytap, ensuring compliance, performance optimization, and security.

B.3.4.1 Azure Monitor

Azure Monitor collects and analyzes telemetry data from Skytap environments, providing real-time insights into infrastructure health and performance. It enables administrators 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 enhance log querying and long-term storage.

B.3.4.2 Azure Functions

Azure Functions acts as a serverless automation layer, enabling event-driven processing of Skytap logs. It supports:

- Automated log forwarding from Skytap to Azure Log Analytics and third-party such as tools SIEM tools, among others.
- Execution of security rules and remediation actions in response to anomalies detected in log data.
- Integration with Azure Logic Apps, enabling workflow automation based on Skytap audit logs.
- IBM i OS log forwarding, allowing organizations to stream logs for centralized monitoring.

B.3.4.3 Azure Log Analytics

Azure Log Analytics processes Skytap audit logs using Kusto Query Language (KQL) for advanced log searches, correlation, and analysis. It allows teams 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 (e.g., virtual machine setup, deletion, and modification) using structured reports.
- Incident response by integrating log analytics with Azure Monitor.

B.3.4.4 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, ensuring alerts are displayed and shared with IT operations teams.

> **Note:** Skytap audit logs integrate with Azure Webhooks and Logic Apps for ingestion and analysis.
>
> - Audit Webhooks send logs to Azure Log Analytics using OpenSSL authentication.
> - Custom pipelines structure IBM Power Systems workload logs for retention.
> - Azure Functions and Monitor alerts detect anomalies and security risks in real time.

B.3.5 Automation

Automation is important in managing IBM Power Systems workloads within Skytap on Azure, allowing for operations, enhanced scalability, and reduced manual effort. Using Azure-native automation tools, Ansible, and Terraform, you can deploy, configure, and manage workloads.

B.3.5.1 Automating operations

Skytap provides a REST API that allows automation of tasks, including VM provisioning, scaling, template creation, and storage management. The following sections outline different automation approaches:

Infrastructure as Code (IaC) with Terraform

Terraform is a declarative automation tool that enables infrastructure provisioning in Skytap on Azure. You can define their Skytap environments as code, ensuring consistency and repeatability. The following key use cases are described:

- 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 is 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 provisions a Skytap environment and deploys an IBM i virtual machine (VM).

- The Skytap provider connects using API credentials.
- A new environment is created with a specified name and description.
- A VM instance is provisioned within this environment using a predefined template ID.
- The VM is allocated vCPUs and memory according to workload requirements.

Note: For further explanation on Infrastructure as Code (IaC) with Terraform for Skytap, visit `Skytap Terraform Provider Documentation`. This resource includes details on the Terraform provider, getting started guides, use cases, and best practices for managing Skytap environments through Terraform.

Automation with Ansible

Ansible provides a declarative, agentless approach to automating Skytap on Azure environments. By leveraging Skytap’s REST API, Ansible can provisioning, configuration, and lifecycle management of IBM i, AIX, and Linux workloads. Below you can see key use cases:

- 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.

As follows are examples showcasing how to integrate Ansible with Skytap for automation.

Controlling the power state of Skytap virtual machines (VMs) ensures efficient resource usage and automation within cloud environments. Example B-2 demonstrates how to start and stop Skytap VMs via API-based automation.

Example: B-2 Start and Stop 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 automates VM lifecycle operations, ensuring power state management while integrating 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 is handled via an API token in the request headers.
- This automation reduces manual effort, ensuring VMs can be controlled programmatically.

Dynamically adjusting CPU and memory resources ensures optimal workload performance and cost efficiency. Example B-3 modifies VM specifications through API requests.

Example: B-3 Scale 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 on page 402 ensures that workloads can dynamically scale to meet demand without manual intervention.

- The API request updates CPU cores and memory allocation for a given Skytap VM.
- The desired 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 uses workload replication, enabling standardized deployments for development, testing, and disaster recovery (DR). Example B-4 creates a Skytap template by cloning an existing environment.

Example: B-4 Create 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 automates template creation, reducing deployment time for recurring environments.

- The API request clones an existing environment, assigning it a template name.
- The template can be used for disaster recovery (DR), testing, and workload replication.
- Versioning and tagging help track changes and optimize storage use.
- Automating template creation enhances agility and repeatability in cloud deployments.

Automating network configurations ensures that Skytap workloads integrate with Azure Virtual Networks (VNet) and hybrid architectures. Example B-5 This playbook provisions a Skytap network and defines a subnet.

Example: B-5 Configure 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 ensures efficient network configuration and integration with Azure services.

- The API request creates a network with a specified subnet.
- Azure Virtual Network (VNet) peering can extend Skytap connectivity across hybrid environments.
- Security controls, including firewall rules and access lists, should be applied post-deployment.
- Automating networking configurations improves scalability and deployment consistency.

Azure Pipelines automating

Users can automate deployment, configuration, and management of IBM i, AIX, and Linux workloads with Azure DevOps Pipelines. Integrating Terraform for provisioning and Ansible for configuration ensures a repeatable, and secure automation framework.

Infrastructure provisioning with Terraform

Azure Pipelines enables the execution of Terraform configurations to provision and manage Skytap environments, including virtual machines (VMs), networks, and templates. By using the Skytap Terraform provider, teams can define infrastructure as code (IaC) and enforce consistency across deployments.

Configuration management with Ansible

Once Terraform provisions Skytap VMs, Azure Pipelines executes Ansible playbooks to automate configuration, software installation, security settings, and workload deployments. This ensures 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:

1. Terraform Deployment Stage
   - Initializes Terraform and applies configurations for Skytap resources.
   - Provisions IBM i, AIX, or Linux VMs in Skytap with predefined specifications.
2. Ansible configuration stage
   - Executes Ansible playbooks to apply system settings and software configurations.
   - Ensures consistency and compliance across Skytap workloads.
3. 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.

B.4 AI with Azure

Enterprises gain a competitive edge by integrating IBM i, AIX, and Linux on IBM Power systems with cloud-based Artificial Intelligence (AI). With Skytap on Azure, you can migrate your mission-critical workloads to the cloud, tapping into Azure AI services for advanced analytics, automation, and insight generation. This approach accelerates Generative AI (GenAI) initiatives and enables real-time, data-driven solutions that were previously challenging due to legacy constraints.

B.4.0.2 Modernizing legacy data for GenAI

GenAI 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 Services, while preserving compatibility with legacy applications. This facilitates Retrieval Augmented Generation (RAG), allowing GenAI queries to reference enterprise-specific data in real time.

Key steps in GenAI adoption with Skytap on Azure include:

- Cloud migration
  - Lift and shift Power workloads to Skytap, reducing physical infrastructure overhead.
  - Integrate with Azure's ExpressRoute or VPN for low-latency and secure connectivity.
- Data preparation
  - Label, format, or anonymize data for GenAI training.
  - Store unstructured content (documents, 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.
  - Ensure data privacy by using secure cloud architecture and Skytap's native audit logs.
- Chat Interface and RAG
  - Deliver GenAI responses via Azure OpenAI chat completion.
  - Deploy semantic search to enhance user queries with enterprise data from Skytap environments.

B.4.0.3 AI Assistant

Skytap's AI Assistant leverages GenAI to optimize resource management, provide contextual insights, and operations. It operates on two knowledge domains:

1. Skytap-specific – Answers queries about Skytap documentation, user guides, and best practices.
2. General knowledge – Provides broad technical information, including industry standards and AI fundamentals.

By integrating *Azure serverless services*, AI Assistant can dynamically scale and deliver real-time recommendations on system performance, user workloads, and troubleshooting. This supports both technical teams, who benefit from in-depth Skytap knowledge, and non-technical teams, who require intuitive cloud operations guidance.

Benefits of AI

The following benefits of AI in Skytap on Azure are described as follows:

- 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; organizations can focus on cloud-enabled analytics.
- Enhanced security

Maintains data governance via Skytap’s native auditing and Azure compliance frameworks.

- Scalable compute

  Uses Azure’s high-performance infrastructure for GenAI, ensuring on-demand resource availability.

Note: By combining Skytap on Azure with *Azure AI services* and *GenAI models*, you can unlock new levels of operational efficiency, market responsiveness, and innovation—while preserving the reliability and performance of the legacy systems.

C

Details for Performance Claims

This appendix provides the technical details behind the performance claims made in the book.

The details are provided for the following claims:

- C.1, “Improved Performance” on page 408
- C.2, “Run AI on a highly performant, sustainable platform” on page 408
- C.3, “Improved Economics” on page 408

© Copyright IBM Corp. 2025.

C.1 Improved Performance

Here are the supporting details for the improved performance of AI on IBM Power discussed in “Improved Performance” on page 44.

Comparison based on IBM internal testing of question and answer inferencing using PrimeQA model[1](based on Dr. Decr and ColBERT models). 42% is based on total throughput in score (inferences) per second on IBM Power S1022 (1x20-core/512GB) running SMT 4 versus Intel Xeon Platinum 8468V-based (1x48-core/512GB) systems. The results are shown in Table 10-2.

Table 10-2 .Test Results

Server	Inferences per second	Concurrent Users
IBM Power S1022	6.26	40
Supermicro SYS-221H-TNR[a]	4.4	40

[a] https://www.supermicro.com/en/products/system/hyper/2u/sys-221h-tnr

- Results valid as of Aug 22, 2023, and conducted under laboratory conditions, individual results can vary based on workload size, use of storage subsystems and other conditions.
- IBM Power S1022 (2x20-core 2.9-4GHz/512GB) used a chip NUMA aligned 10-core LPAR
- Test was run with Python and Anaconda environments including packages of Python 3.9 and PyTorch 2.0. The Python libraries used are platform-optimized for both Power and Intel.
- Batch size = 60 with 40 concurrent users.
- The torch.set_num_threads(int) optimized across a variety of load levels.
- Models fine-tuned by IBM on a corpus of IBM-internal data

C.2 Run AI on a highly performant, sustainable platform

Here are the supporting details for the improved performance of AI on IBM Power discussed in “Run AI on a highly performant, sustainable platform” on page 44.

- Based on IBM internal testing of data science components, (WML, WSL, Analytic Engine) of Cloud Pak for Data version 4.8 in OpenShift 4.12.
  - Results valid as of 11/17/2023 and conducted under laboratory condition.
  - Individual results can vary based on workload size, use of storage subsystems & other conditions.

C.3 Improved Economics

Here are the supporting details for the improved performance of AI on IBM Power discussed in “Improved Economics” on page 44.

1. Based on IBM internal testing of data science components, (WML, WSL, Analytic Engine) of Cloud Pak for Data version 4.8 in OpenShift 4.12. Results valid as of 11/17/2023 and conducted under laboratory condition. Individual results can vary based on workload size, use of storage subsystems & other conditions.

[1] https://github.com/primeqa

2. The workload mimics a real-time fraud detection logic flow. JMeter is used to submit credit card transactions for different user id and card number combinations. The inferencing application running as microservices in Cloud Pak for Data deployment space extracts the user id and credit card number and uses them to look up 6 previous transactions of the same user and card combination from the Db2 database which is also running within the Cloud Pak for Data cluster. The data retrieved from the database is then combined with the new entry and pass to the LSTM model to determine whether the latest transaction is fraud or not. The score (value between 0 to 1) is returned to the JMeter client as an indicator of whether that transaction is likely a fraud or not.
3. The measurement used for both Power and Intel systems is the throughput result (score/second) reported by JMeter, when running 192 current threads (1 thread representing 1 user) against 96 inferencing end points.
4. Power10 S1022 has a total of 40 physical cores and 2 TB RAM (machine type 9105-22A).
5. There are 7 LPARs on this system including 3 master nodes of 2 cores and 32 GB RAM each, 3 worker nodes of 10 cores and 490 GB RAM each, and a bastion node of 4 cores 128 GB RAM. Local 800 GB NVME drives are used as boot drives for each node, and one 1.6TB NVMe is used for NFS server storage running on the bastion node. There is one 100G Ethernet adapters virtualized through SRIOV, with each LPAR taken 10% of network bandwidth. Each LPAR ran with CPU frequency range 3.20GHz to 4.0GHz. All 3 worker nodes ran in SMT 4 mode, while master and bastion nodes ran in SMT 8 mode.
6. The Intel system is Xeon Platinum 8468V with 96 physical cores and 2 TB RAM.
7. The KVM host takes 2 core and 32 GB RAM, which supports 7 KVM guests on this system, including 3 master nodes of 4 cores and 32 GB RAM each, 3 worker nodes of 24 cores and 490 GB RAM each, and a bastion node of 4 cores 128 GB RAM. Local 1.6 GB NVME drives are used as boot drives for these nodes, and one 1.6TB NVMe used for NFS storage on the bastion node. There is one 100G Ethernet adapters virtualized through SRIOV. Each KVM guest ran with CPU frequency range from 2.40GHz to 3.8GHz. All nodes are RHEL CoreOS KVM guests running on the server with hyperthreading enabled.
8. Pricing is based on:
   - Power S1022[2].
   - Typical industry standard Intel x86 pricing[3]
   - IBM software pricing[4]

   Add text here at list 2 level (Body2)

   - Add list text here (ListBulleted 3)

     Add text here at list 3 level (Body3)

[2] `https://www.ibm.com/power/pricing/us-en#oracle`
[3] https://www.synnexcorp.com/us/govsolv/pricing/
[4] https://www.ibm.com/downloads/cas/DLBOWBPK

Related publications

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

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *Introduction to IBM PowerVM*, SG24-8535
- *IBM Power E1080 Technical Overview and Introduction*, REDP-5649
- *IBM Power E1050 Technical Overview and Introduction*, REDP-5684
- *IBM Power 10 Scale Out Servers Technical Overview S1012, S1014, S1022s, S1022 and S1024*, REDP-5675
- *Using Ansible for Automation in IBM Power Environments*, SG24-8551
- *Simplify Management of IT Security and Compliance with IBM PowerSC in Cloud and Virtualized Environments*, SG24-8082
- *IBM Power Systems SR-IOV: Technical Overview and Introduction*, REDP-5065
- *IBM Storage Scale Information Lifecycle Management Policies*, REDP-5739
- *IBM Storage Scale System Introduction Guide (ESS)*, REDP-5729
- *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
- *IBM Storage Ceph Concepts and Architecture Guide*, REDP-5721
- *IBM Storage Fusion Product Guide*, REDP-5688

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources

- What is Kubernetes?

  https//www.ibm.com/think/topics/kubernetes
- What is Docker?

  https//www.ibm.com/think/topics/docker

© Copyright IBM Corp. 2025.

- What is Podman?

  https//www.redhat.com/en/topics/containers/what-is-podmanv

- 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

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

IBM

(1.5” spine)
1.5”<-> 1.998”
789 <->1051 pages

Redbooks

Modernization Techniques for IBM Power

SG24-8582-00

ISBN

IBM

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

Redbooks

Modernization on Power

SG24-8582-00

ISBN

IBM

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

Redbooks

Modernization on Power

(0.2”spine)
0.17”<->0.473”
90<->249 pages

(0.1”spine)
0.1”<->0.169”
53<->89 pages

Redbooks

Modernization Techniques for IBM Power

SG24-8582-00

ISBN

IBM

(2.5" spine)
2.5"<->nnn.n"
1315<-> nnnn pages

Redbooks

Modernization Techniques for IBM Power

SG24-8582-00

ISBN

IBM

(2.0" spine)
2.0" <-> 2.498"
1052 <-> 1314 pages

SG24-8582-00

ISBN

Printed in U.S.A.

Get connected

ibm.com/redbooks