ibm.com/redbooks

Introduction to Deployable Architectures in IBM Power Virtual Server

Tim Simon

Srikanth Joshi

Adhish Kapoor

Raydo Matthee

Osman Omer

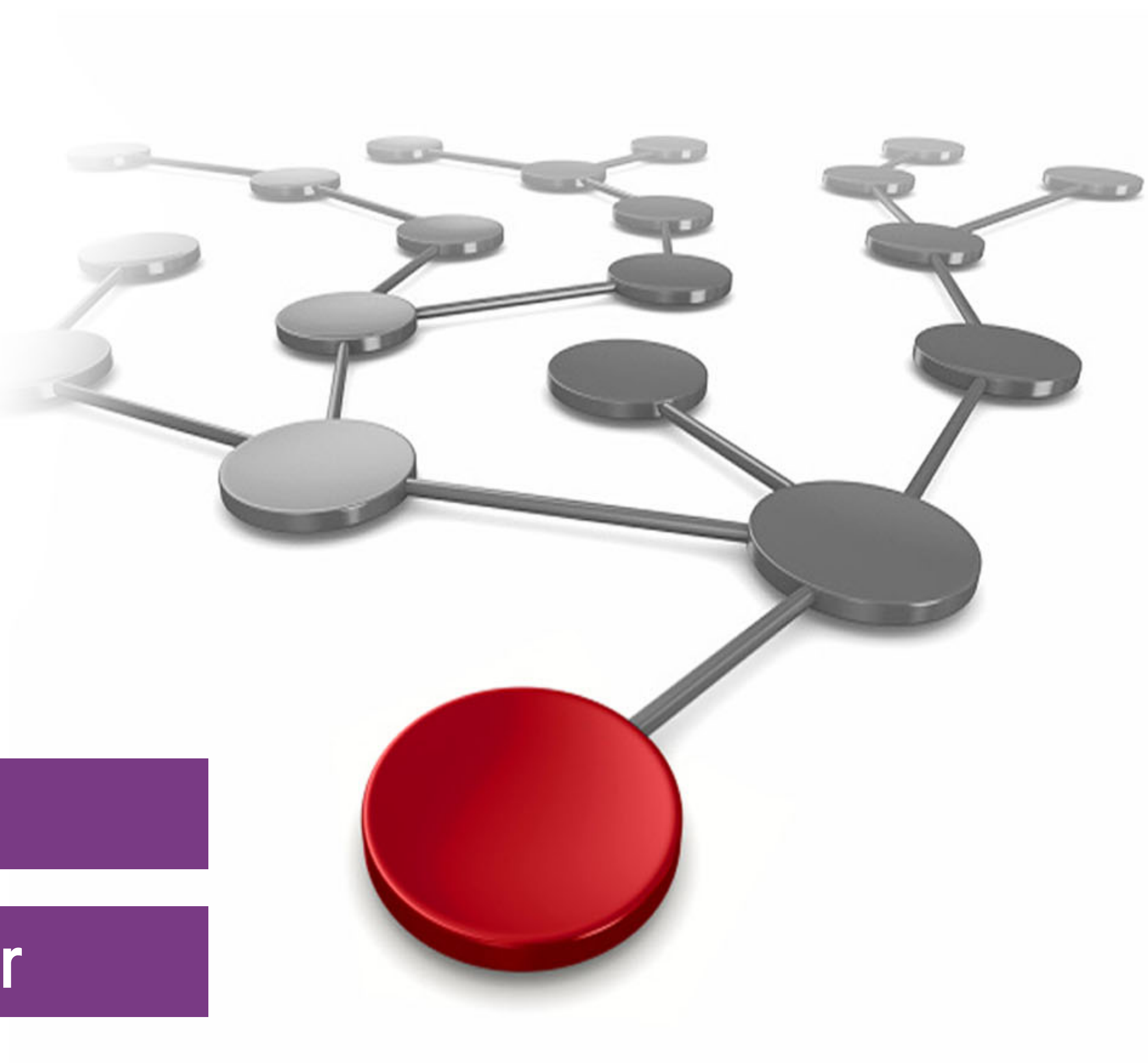
Jose Pereira

Syed Maaz Siddique

Gary Smith

Guido Somers

Henry Vo



Cloud

IBM Power

Redbooks

IBM Redbooks

Power Virtual Server Deployable Architecture

February 2025

SG24-8576-00

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

First Edition (February 2025)

This edition applies to:
Power Virtual Server (December 2024 release)
SUSE Linux Enterprise Server 15 SP 3 for SAP
SUSE Linux Enterprise Server 15 SP 4 for SAP
Red Hat Enterprise Linux 8.4 for SAP
Red Hat Enterprise Linux 8.6 for SAP

© 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®
IBM®
IBM Cloud®
IBM Cloud for Financial Services®
IBM Cloud Satellite®
IBM FlashSystem®
Maximo®
Orchestrate®
POWER®
Power9®
PowerHA®
PowerVM®
Redbooks®
Redbooks (logo) ®
Satellite™
System z®

The following terms are trademarks of other companies:

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

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

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

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

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

Preface

IBM Cloud's deployable architectures provide standardized patterns for deploying and managing applications and infrastructure. They accelerate deployment, enhance efficiency, and improve security and compliance when creating complex environments. These architectures offer a ready-to-use blueprint with pre-configured components and best practices, streamlining the deployment process and minimizing manual intervention. They also incorporate security best practices and comply with industry regulations, ensuring a secure and compliant environment. By using pre-defined configurations, organizations can optimize their cloud spending with efficient resource utilization.

In addition to pre-defined architectures, IBM Cloud provides tools and services to create custom deployable architectures. This enables you to simplify and accelerate cloud deployments while ensuring secure, compliant, and cost-effective solutions.

This IBM Redbooks publication provides a how-to usage content perspective, covering deployment, networking, and data management tasks to help you implement deployable architectures in Power Systems Virtual Server. It addresses topics for IT architects, IT specialists, developers, sellers, and anyone implementing and managing workloads in Power Systems Virtual Server. The publication also covers transferring skills to technical teams and solution guidance to sales teams.

This book complements the IBM Documentation web page and aligns with IBM Garage for Systems Technical Education's educational materials.

Authors

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

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

Srikanth Joshi is an Power Virtual Server Private Cloud Test Architect at IBM India Systems Development Lab in Bangalore, India, with over 16 years of experience in the information technology field. His areas of expertise are IBM AIX® Live Kernel Update, IBM PowerVM® concepts like Virtual IO Server, Shared Storage Pools, N-port ID Visualization, SRIOV, Live Partition Mobility, System management products like HMC, PowerVC, IBM Cloud® services Power Virtual Server, Cloud Object Storage etc. He has also performed various other roles like technical lead, release management and project management. He has published couple of community blogs in the area of AIX HPCS (HyperProtect Crypto Services) and Power Virtual Server along with developer work articles. He has also given technical sessions to System Integrators and helped them to enable latest IBM POWER® technologies. He can be reachable via mail srikanth.joshi@in.ibm.com.

Adhish Kapoor is a team lead for the VMRM HA/DR product, with over 20 years of experience and expertise in designing and developing High Availability (HA) solutions,

© Copyright IBM Corp. 2025.

Disaster Recovery (DR) systems, storage replication, parallel network file systems (pNFS), deduplication products, and container technologies. Additionally, Adhish has worked to deliver robust and scalable solutions for complex data center configurations.

Raydo Matthee is a technology expert specializing in IBM Power Systems and cloud infrastructure solutions. As a co-author of Introduction to Deployable Architectures in IBM Power Virtual Server, he has played a pivotal role in developing strategies for secure and scalable enterprise deployments. With a deep understanding of cloud architecture, high-availability systems, and technical training, Raydo brings a practical and forward-thinking approach to solving complex IT challenges. His contributions to IBM Redbooks highlight his dedication to innovation and excellence in enterprise technology.

Osman Omer is a senior IT Managing Consultant based in Qatar. He has worked for IBM for 20 years. He worked as a software engineer for 8 years in Rochester, Minnesota before joining Lab Services. He has worked for IBM Systems management, cloud solutions, and automation services. His first project was porting IBM i to be managed by HMC, and then worked on IBM i OS enable system management, tools, Systems Director, VMControl, and PowerVC. As a Lab Services consultant, he helps IBM customers with the products that he used to develop. After moving to Qatar, Osman became a member of the MEA team that is responsible for cloud and automation services delivery in the region. In 2023 he acted as the EMEA Power Services Delivery Practice Leader in addition to his consulting and leadership responsibilities. As of 2024, he expanded his roles and responsibilities into Technical Account Management for major banks in Europe and MEA, which is an important role to help these banks maximize their benefits of the Power Systems platforms. Osman holds a master degree in Computer Science from South Dakota State University.

Jose Pereira is a Senior IBM Public Cloud Technical Specialist for Canada Technical Sales. He specializes in designing and implementing scalable infrastructure solutions on IBM Cloud. Through strategic implementation of IBM Cloud services, including containerization, automation, serverless computing, and hybrid cloud architectures, he creates resilient solutions that enable business growth and digital transformation. Outside of work, he's an avid comic book collector who enjoys discovering the next great addition to his collection.

Syed Maaz Siddique is an Infrastructure Architect and Site Reliability Engineer (SRE) at Kyndryl Pvt Ltd in India, with over 16 years of experience in the information technology field, including more than 12 years at IBM and Kyndryl. He holds a Bachelor of Engineering in Electronics and Communication from Oxford College. Maaz specializes in AIX and UNIX platforms, including Red Hat Enterprise Linux (RHEL), Solaris, and SUSE Linux Enterprise Server (SLES). He is an IBM Cloud Certified Professional Architect and a Kyndryl SRE Level 1 Certified Expert, focusing on cloud solutions and enhancing system reliability. His expertise also encompasses virtualization technologies like PowerVM, IBM PowerHA®, Oracle and Red Hat virtualization, as well as data center migration and disaster recovery strategies. Additionally, he is skilled in automation and management tools, including Red Hat IBM Satellite, Red Hat Ansible, and Red Hat OpenShift, making him an asset in driving innovation and operational excellence within the IT landscape.

Gary Smith is an IBM Cognitive Power Systems Topgun in Houston, Texas with over 40 years of experience in service, support and training on IBM Power, Storage Solutions, Z, and software platforms. Gary has experience in product service plans, education development and delivery for IBM Power solutions. His background is in developing and writing education packages and product installation and service documentation and additionally he has worked on product quality with IBM Development organizations.

Guido Somers is a Distinguished Technical Specialist working at IBM Switzerland. He has almost 30 years of experience in the information technology field, 26 years of which were with IBM. He holds several degrees, and he did research in the field of theoretical physics. His areas of expertise include cloud, IT architecture, virtualization, cross-platform infrastructure, IT analytics, ISV Solutions - SAP, Oracle, and Cisco, Blockchain, data center migration and optimization, automation, artificial intelligence (AI), and security. He is an IBM Redbooks Platinum author.

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¬Æ 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.

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

1

Introduction to Deployable Architectures

IBM Cloud empowers organizations to modernize applications, improve data management, and accelerate innovation by offering a comprehensive solution for businesses. This versatile cloud platform provides both Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) solutions, catering to businesses of all sizes. It delivers a secure, scalable, and globally accessible environment for deploying and managing applications. IBM's Power Virtual Server, an IaaS solution, enables rapid deployment of virtual servers on IBM Power hardware, ideal for migrating AIX, IBM i, and Linux workloads to the cloud. IBM supports Deployable Architectures to assist in implementing and managing both IBM Cloud and Power Virtual Server. These automation templates streamline infrastructure deployment. This chapter provides a high-level overview of IBM Cloud and IBM Power Virtual Server, introducing Deployable Architectures and demonstrating how automation simplifies the implementation of complex application environments like SAP HANA.

This chapter contains the following topics:

© Copyright IBM Corp. 2025.

1.1 IBM Cloud introduction

The IBM Cloud platform combines platform as a service (PaaS) with infrastructure as a service (IaaS) to provide an integrated experience. The platform scales and supports both small development teams and organizations, and large enterprise businesses. Globally deployed across data centers around the world, the solution you build on IBM Cloud spins up fast and performs reliably in a tested and supported environment you can trust!

IBM Cloud provides solutions that enable higher levels of compliance, security, and management, with proven architecture patterns and methods for rapid delivery for running mission-critical workloads. Available in data centers worldwide, with multizone regions in North and South America, Europe, Asia, and Australia, you are enabled to deploy locally with global scalability.

IBM Cloud offers the most open and secure public cloud for business with a next-generation hybrid cloud platform, advanced data and AI capabilities, and deep enterprise expertise across 20 industries. Solutions are available depending on your needs for working in the public cloud, on-premises, or a combination:

- With public cloud, the resources are made available to you over the public internet. It is a multi-tenant environment, and resources like hardware and infrastructure are managed by IBM.
- A hybrid cloud solution is a combination of public and private giving you the flexibility to move workloads between the two based on your business and technological needs. IBM uses Red Hat OpenShift on IBM Cloud, the market-leading hybrid cloud container platform for hybrid solutions that enables you to build once and deploy anywhere. With IBM Cloud Satellite®, you can create a hybrid environment that brings the scalability and on-demand flexibility of public cloud services to the applications and data that runs in your secure private cloud.
- Support for multicloud and hybrid multicloud solutions is also available, which makes it easy for you to work with different vendors. IBM Cloud Paks are software products for hybrid clouds that enable you to develop apps once and deploy them anywhere.
- Virtual Private Cloud (VPC) is available as a public cloud service that lets you establish your own private cloud-like computing environment on shared public cloud infrastructure. With VPC, enterprises can define and control a virtual network that is logically isolated from all other public cloud tenants, creating a private, secure place on the public cloud.

With our open-source technologies, such as Kubernetes, Red Hat OpenShift, and a full range of compute options, including virtual machines, containers, bare metal, and serverless, you have the control and flexibility that's required to support workloads in your hybrid environment. You can deploy cloud-native apps while also ensuring workload portability.

Whether you need to migrate apps to the cloud, modernize your existing apps by using cloud services, ensure data resiliency against regional failure, or use new paradigms and deployment topologies to innovate and build your cloud-native apps, the platform's open architecture is built to accommodate your use case.

1.2 Power Virtual Server overview

IBM Power Virtual Server is an offering from IBM that allows you to deploy a virtual server – also known as a logical partition (LPAR) – in just minutes. This solution provides flexible, secure, and scalable compute capacity for enterprise workloads running on IBM Power. It provides an excellent platform for moving AIX, IBM i, and Linux on Power workloads to the cloud without a time consuming and risky migration of those workloads to a different platform.

With fast and flexible management, you can connect to a range of enterprise services from IBM, all under a pay-as-you-use billing model. This enables you to easily adjust workloads and optimize compute capacity. Power Virtual Server instances are available globally across the IBM Cloud platform, allowing you to quickly deploy resources tailored to your business needs and efficiently manage workload demands. Power Virtual Server allows enterprises to minimize capital expenses and mitigate risks when migrating critical workloads to the cloud.

Customers can provision flexible, secure, and scalable compute capacity for IBM Power based enterprise workloads either off-premises (in an IBM data center), or on-premises in your own data center in a cloud environment with no upfront costs and a pay-as-you-go pricing mode. Whether you choose Power Virtual Server off-premises or on-premises, you utilize the same IBM management tools to order and manage your environment.

Power Virtual Server is a single offering that can be delivered in two variations; off-premises, where the infrastructure components are located in IBM data centers, and on-premises, where the infrastructure components are located in the client's data center. The on-premises variation is also known as IBM Power Virtual Server Private Cloud. Both variations provide a cloud-based consumption model, where you pay for resources as they are consumed, and are managed with the same management interfaces.

Figure 1-1 shows the two different implementations of IBM Power Virtual Server.

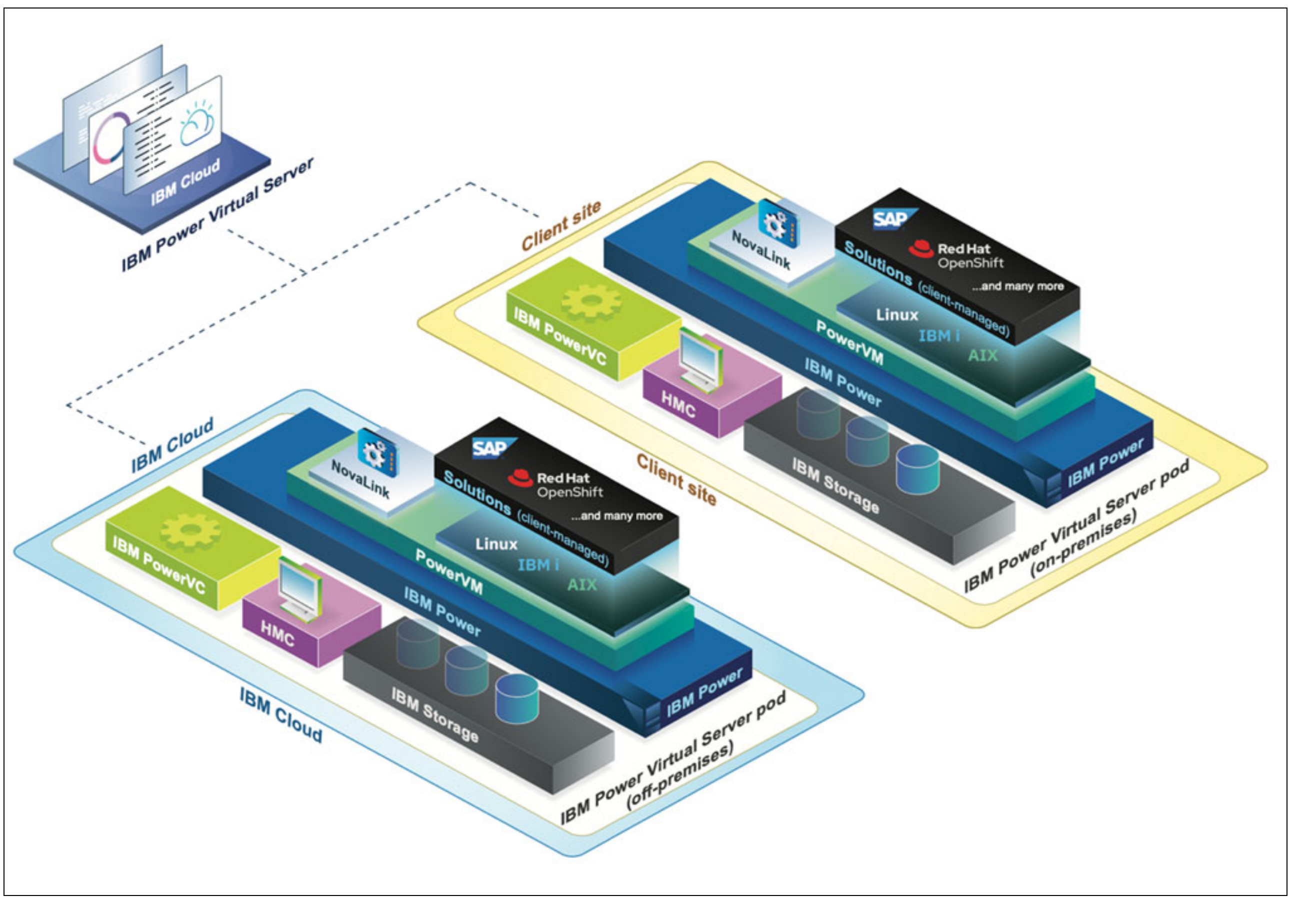


Figure 1-1 Power Virtual Server variations

Whether using the on-premises Power Virtual Server Private Cloud offering or the off-premises public cloud offering, creating your environment starts with creating an IBM Cloud account. Figure 1-2 will shows the steps to get started with Power Virtual Server – off-premises or on-premises.

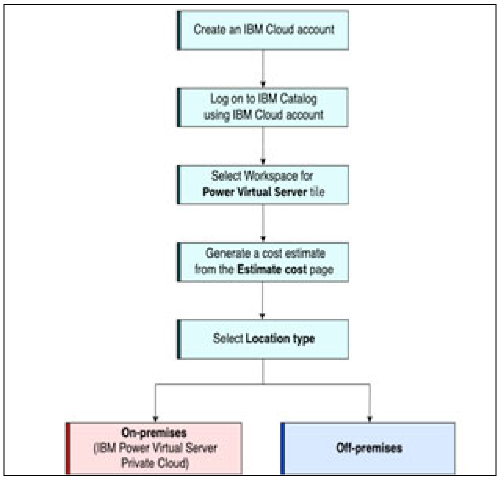


Figure 1-2 How to get started with Power Virtual Server

1.2.1 IBM Power Virtual Server located in IBM Data centers (Off-premises)

In this offering, IBM Power Virtual Server resources reside in IBM data centers with dedicated networking and storage area network (SAN) attached storage. Customers can choose from one of the currently 21 different data centers in the location list, choosing the data center that is closest to their users. IBM Power clients who rely on private cloud infrastructure can now quickly and economically extend their Power IT resources on the cloud.

In these IBM data centers, Power Virtual Server are separated from the rest of IBM cloud servers with separate networks and direct attached storage. In this offering, customers are given flexibility to choose from different IBM Power server models, an IBM supplied OS image, and different tiers for storage based on your specific workload requirements. Customers are also given to choose their own image as Bring Your Own License (BYOL) so that they can bring already configured images from their enterprise infrastructure onto Power Virtual Server. This offering also supports distinctive features such as shared processor pools, public connectivity of VMs, placement groups, global replication service (GRS) which can be consumed based on customer's requirements. Figure 1-3 shows the workflow to starting your implementation of IBM Power Virtual Server off-premises.

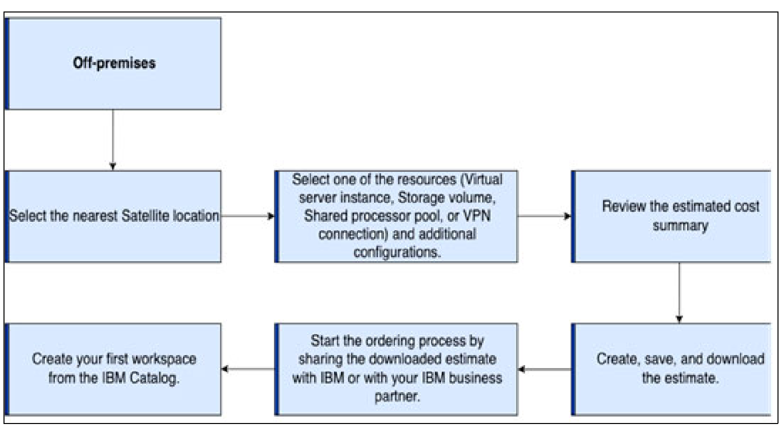


Figure 1-3 Workflow to implement Power Virtual Server public cloud

1.2.2 IBM Power Virtual Server Private Cloud (On-premises)

IBM Power Virtual Server Private Cloud was created as part of IBM's distributed hybrid infrastructure strategy. Power Virtual Server Private Cloud extends all the benefits of IBM Power Virtual Server into your (or a partner's) data center. The enhanced capabilities of IBM Power Virtual Server Private Cloud provide managed infrastructure as a service at client locations, with metered consumption and no upfront costs to support Hybrid by Design delivery of services.

The IBM Power Virtual Server Private Cloud offering is engineered to:

- Maintain customer data and workloads on your own site.

  Enterprises may have workloads or data that is regulated and cannot be hosted off-premises. In some cases, enterprises can have workloads that are sensitive or with ultra-short latency requirements that are better served on site and in very close in proximity with other on-site workloads.

- Maintain customer data in region and specific geographies in the location of their choice.

  Country sovereignty regulations are requiring some data and workloads to stay in country. According to a recent IBM Institute of Business Value study, 61% of cloud leaders cite security or compliance as reasons for moving certain workloads from public clouds to private clouds or on-premises data centers.

- Provide a seamless hybrid cloud experience.

  Enterprises can foster a unified hybrid cloud landscape by seamlessly integrating Power Virtual Server running both at an IBM site and at a client site location with the ability to manage all the virtual machines (VMs) and infrastructure effortlessly through a unified user interface. Clients can receive the flexibility utilizing as-a-service with intentional workload placement on and off premises.

- Deliver predictable charging model with committed monthly spend combined with flexible consumption with metered usage-based pricing.

  Both Power Virtual Server offerings, off-premises running at an IBM location and on-premises running at the client site, include compute, memory, storage, and operating system licenses that are fully metered by the hour allowing clients to pay for how much they use each month with no upfront payment.

- Streamline IT operations.

  Whether in the cloud or at an enterprise's site, IBM manages the infrastructure, freeing enterprises to focus on business outcomes and less on managing infrastructure. IBM will own, deliver, and set up the Power Virtual Server in datacenter of choice, and provide a fully managed solution, including monitoring, security, firmware updates, and infrastructure management.

- Provide enhanced security and control of data.

  IBM Power Virtual Server is designed to provide comprehensive security for IBM Power infrastructure by integrating with IBM Cloud tooling to manage security. This alleviates the need to manage Power infrastructure security with the added benefit of maintaining sensitive data and workload on-premises.

The physical infrastructure is delivered as a point of delivery (pod) which will be deployed in customer's data center. A pod is the physical component which resides within the client datacenter and contains the compute, storage an,d network components. A pod contains one or more racks where each of the components are installed. The racks are interconnected to provide a completely self-contained infrastructure, including both customer usable components, spare components, and management components.

This pod will be maintained by IBM site reliability engineering (SREs) and managed through IBM Cloud platform. Each pod is associated with an IBM Cloud satellite location that is owned by customer's IBM cloud account. This architecture provides the ability to scale your private cloud infrastructure horizontally by adding more pods to meet your workloads requirements.

Figure 1-4 shows the workflow involved in implementing the on-premises (Private Cloud) capabilities of Power Virtual Server.

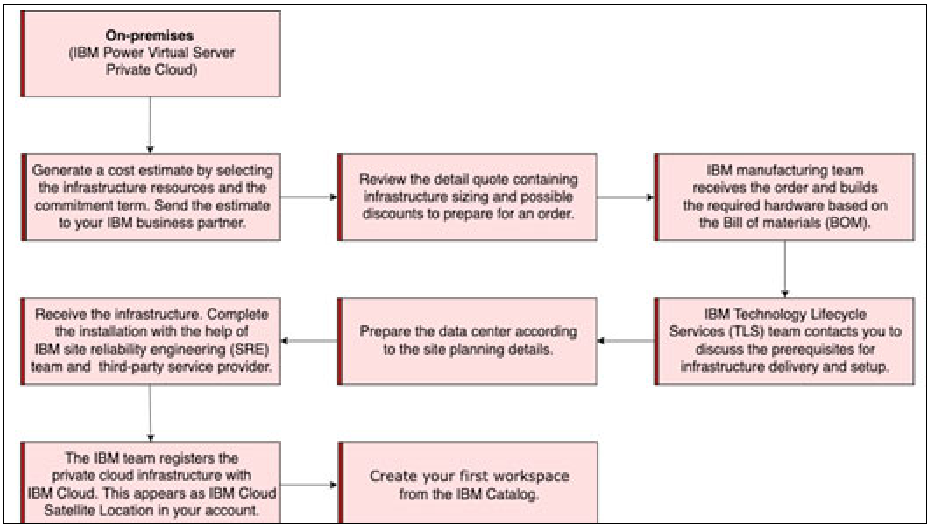


Figure 1-4 Power Virtual Server Private Cloud (on-premises) workflow

1.3 What is a Deployable Architecture

A deployable architecture provides cloud automation functions through a composable infrastructure. It combines one or more cloud resources to provide a common architectural pattern. It is designed for simplified deployment by users, scalability, and modularity, allowing users to easily provision and manage infrastructure resources. A deployable architecture incorporates one or more modules. Deployable architectures are coded in Terraform, which are configured with input variables to achieve the behavior that is required. Deployable architectures can also claim compliance with security frameworks and controls.

This solution can be deployed from the IBM Cloud catalog. Before its deployment you can see the cost estimates of each variation. Another possibility is to share your own deployable architecture within your enterprise in a private catalog. As such, it can help you to create a framework around how resources are deployed within your enterprise. By working with these reusable configurations, you can define the standard for deployment once and ensure that it is easily repeatable throughout the company.

Rather than spending months figuring out how to get services and software to work together, you can use IBM Cloud's well-architected patterns. Each pattern is packaged as composable, automated building blocks known as modules and deployable architectures. A module is a stand-alone unit of automation code that can be reused and shared as part of a larger system. Although these modules can be used alone, they're more powerful when combining them to build a deployable architecture. Modules that are created by IBM Cloud are made available in the IBM

Terraform modules public GitHub organization. Figure 1-5 is a high level view of a deployable architecture.

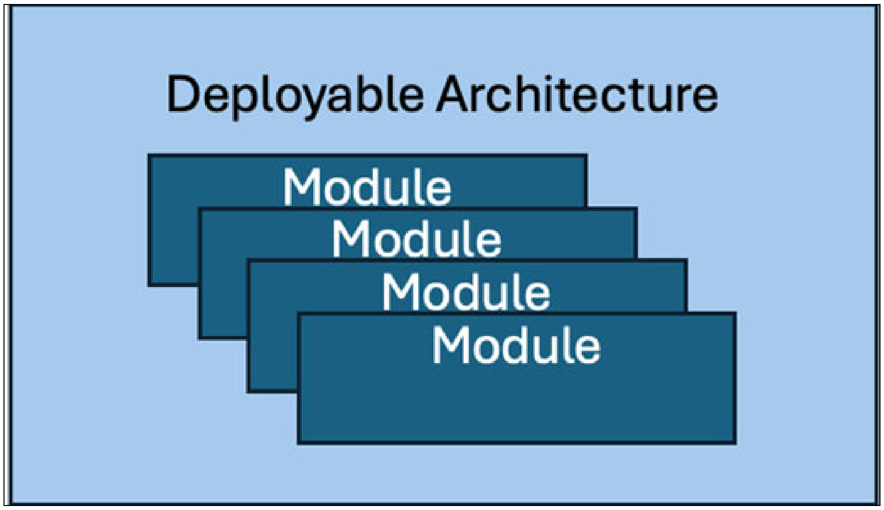


Figure 1-5 Deployable Architecture and Modules

A deployable architecture might include variations as well as have dependencies. A variation is a type of deployable architecture that applies differing capabilities or complexity to an existing deployable architecture. For example:

– A *Quick start* variation that has basic capabilities for simple, low-cost deployment to test internally.

– A *Standard* variation that is a bit more complex that is ready for use in production.

When it comes to dependencies, a deployable architecture is considered dependent upon another architecture when it has inputs that require the outputs from that other architecture to properly deploy.

1.3.1 Benefits of automation

Cloud computing has revolutionized the way applications and infrastructure are built, deployed, and managed. A deployable architecture becomes even more important in cloud environments due to the dynamic nature of resources and the scalability cloud platforms offer. Here is why deployable architectures and automation is essential in the cloud:

- Cloud-Native Advantages

  Cloud environments are designed to be dynamic, scalable, and resilient. A deployable architecture aligns with these principles by enabling seamless integration with cloud services. Features like automated scaling, load balancing, and global distribution are fully leveraged when the architecture is optimized for deployment in the cloud.

- Elasticity and On-Demand Scalability

  Cloud platforms allow you to scale resources up or down based on demand. A deployable architecture ensures that this scaling is seamless and cost-efficient.

  – Horizontal Scaling: Add more instances to handle traffic spikes.
  – Vertical Scaling: Upgrade the resources of existing instances when necessary.

- Multi-Environment Consistency

  Cloud environments often include multiple stages: development, testing, and production. A deployable architecture ensures these environments remain consistent, which is critical for avoiding "it works locally, but not in production" scenarios.

  Infrastructure as Code (IaC) tools like Terraform allow you to define and replicate cloud environments. For example, using a Terraform script to provision identical environments for QA and production on cloud environment ensures there are no discrepancies in server configurations or networking setups.

- Cost Optimization

  Cloud providers charge based on resource usage (e.g., compute hours, storage, network). Deployable architecture helps optimize costs by automating the provisioning and decommissioning of resources as needed.

- DevOps benefit from automation

  Continuous Integration/Continuous Deployment (CI/CD) pipelines help automate the process of testing and deploying changes to cloud-based systems.

Automation drives the deployment of a deployable architecture. By scripting routine tasks into modules, the need for manual operations is reduced. A module provides reusable, curated automation to speed up the process for those who are creating a library of shareable automation components. Figure 1-6 shows how modules are used in a deployable architecture.

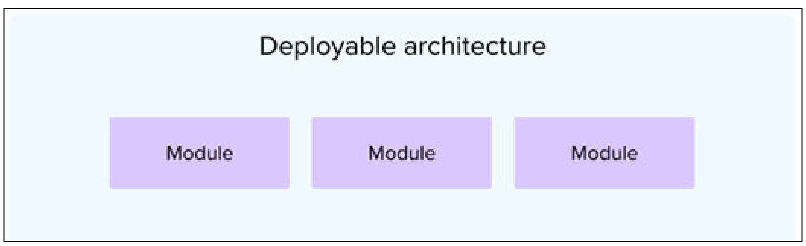


Figure 1-6 Deployable Architecture

The result is consistent, reliable operations that spans various aspects of the architecture from infrastructure provisioning to application deployment and system management. In general, a deployable architecture utilizes tools, methodologies, and best practices that builds a solution framework. For an example, the *Power Virtual Server with VPC landing zone* deployable architecture makes use of the Terraform IBM Module - VPC Landing Zone.[1]

1.3.2 Automation methods and tools

Using Infrastructure as Code (IaC) to provision infrastructure through machine-readable definition files such as Terraform, Ansible, and IBM Cloud Schematics allows teams to define infrastructure in code, enabling consistent and repeatable deployments.

Configuration Management facilitates the automation of virtual server configurations, networking, and storage setups, ensuring that environments are consistent across all stages; development, testing, and production. Deployment Automation using Continuous Integration/Continuous Deployment (CI/CD) with CI/CD pipelines automates the building, testing, and deployment of applications. This accelerates the software delivery life cycle and improves reliability.

[1] https://github.com/terraform-ibm-modules/terraform-ibm-landing-zone

IBM Cloud utilizes both projects and deployable architectures:

- Projects are a named collection of configurations that are used to manage related resources and IaC deployments.
- A deployable architecture is a cloud automation for deploying a common architectural pattern that combines one or more cloud resources that are designed for easy deployment, scalability and modularity.

Projects and deployable architectures enable teams to turn best practices into reusable patterns in order to create new environments in a few clicks and ensure that the environments remain compliant, secure and up to date over time. Deployable architectures can be found in the IBM Cloud catalog, where there are several deployment options available.

1.3.3 Structure of a deployable architecture

You can manually create and customize a deployable architecture using Terraform, or you can download the code from an existing deployable architecture, modify it to fit your needs, and create a new deployable architecture with your changes. For more information see Creating a deployable architecture from an existing offering in IBM Cloud documentation. Figure 1-7 shows the structure of a deployable architecture.

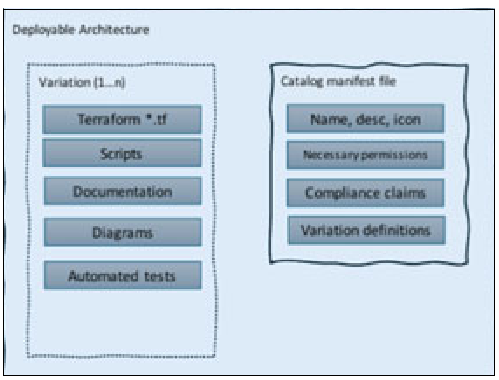


Figure 1-7 Anatomy of a terraform-based deployable architecture

These components are:

- Terraform is used to define the desired end-state of the infrastructure and perform the necessary API requests to create, update, or delete resources to reach that state.
- Optional scripts can be used as a temporary solution for missing functions, such as using Bash or Python, or for ad hoc operational tasks, such as those handled by Ansible.
- Automated tests validate the processes used to deploy, verify, and destroy infrastructure.
- Supporting documentation, such as an architecture diagram and a README file

The catalog manifest is included to:

- Define how deployable architectures appear in the IBM Cloud catalog, including general details (name, description, features),
- Define links to Terraform configurations,
- Define the compliance requirements verified by IBM Cloud Security and Compliance Center, and necessary IAM permissions.
- Deployable architectures can have variations. For example, one can be a basic quick start and a second variation provide a more complex production-ready version. Each variation is on boarded to the catalog from different working directories within the same repository and defined in the ibm_catalog.json file.

For more information on the catalog manifest, see Locally editing the catalog manifest.

1.3.4 Infrastructure as Code

Consider, if instead of manually setting up your infrastructure for production or testing environment, write a code and execution of that code can use to automatically define, deploy, update and destroy your infrastructure. This is the basis idea behind Infrastructure as code.

Traditionally, deployment of any software product requires a lot of manual process like:

- You need to configure Servers.
- Setup appropriate OS version on the servers.
- Networking setups per requirements.
- Deploy required software's on each machine.

The steps listed above are for a simple deployment. For enterprise customers, environment, these steps can be much more complicated and varies as per application requirement. And software product which need to support scalable environment, these deployments get more challenging as machine number can goes up to thousands. Manually configuring these deployments always painful and error prone as per human errors.

Infrastructure as Code (IaC) addresses all these change and helps to deploy these environments automatically with help of code. This marks a significant shift in perspective, where all facets of operations are treated as software – even tasks traditionally tied to hardware, such as configuring physical servers. A fundamental principle of DevOps is that nearly everything can be managed through code, including servers, databases, networks, log files, application configurations, documentation, automated testing, deployment processes, and more.

As a practical example, let us say your company needs 5 servers with 16GB RAM and 500GB storage which will be connected to a database and running a web application.

Instead of setting this up manually, you write a file like Example 1-1.

Example 1-1 Example of infrastructure as code

```
Yaml-> code
resources:
  - server:
      count: 5
      ram: 16GB
      storage: 500GB
      application: "webapp"
  - database:
      type: PostgreSQL
      version: 13
```

When you run this code using an IaC tool, it will create the servers and configure the database for you.

Benefits of Infrastructure as Code

1. Speed and Efficiency

   Automated scripts can set up infrastructure in minutes, something that might take days manually. You can create multiple environments (e.g., for testing or production) in no time.

2. Consistency

   Every environment created with the same code is identical. This eliminates the “it works on my machine” problem.

3. Version Control

   Infrastructure can be stored in version control systems like Git, allowing you to track changes, collaborate with teams, and roll back if something goes wrong.

4. Scalability

   You can scale up resources by modifying a few lines of code. For example, if traffic to your application grows, just increase the number of servers in the configuration file.

5. Cost Savings

   By automating infrastructure, you reduce human effort and prevent errors that could lead to costly downtime.

1.3.5 Using projects for Infrastructure as Code deployments

The IBM Cloud Project is your central dashboard to configure, deploy, and manage the deployable architecture[2]. Figure 1-8 shows a workflow for a deployable architecture project.

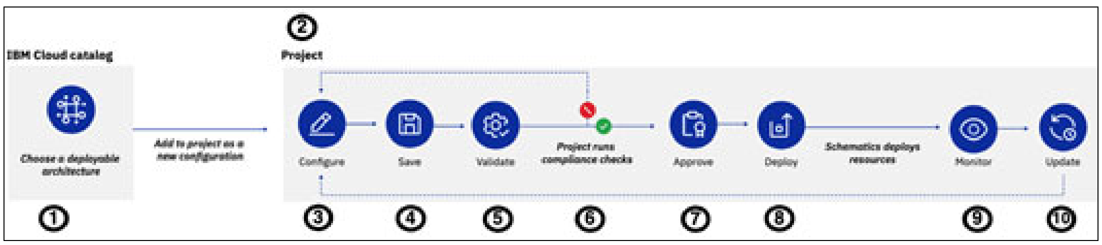


Figure 1-8 Understanding the projects workflow

Figure 1-8 illustrates the workflow for IBM Cloud Infrastructure as Code (IaC) projects, specifically showcasing how resources are deployed and managed through the IBM Cloud platform. It aligns with the process of securely managing configurations and deployments in an enterprise setting using IaC principles. Below is a step-by-step breakdown of the workflow:

1. Choose a Deployable Architecture from the IBM Cloud Catalog

   Select pre-configured or customizable deployable architectures from the IBM Cloud catalog. These architectures define resource types, relationships, and configurations needed for their specific project requirements.

2. Add to Project as a New Configuration

   The selected architecture is added to a project within the IBM Cloud environment, where it becomes a configurable entity.

3. Configure

   Tailor the configuration to meet specific requirements, such as resource types, quotas, region preferences, and security policies.

4. Save

   Configurations are saved to ensure a record of the defined architecture and any customization applied.

5. Validate

   The project performs validation checks to ensure that configurations adhere to predefined standards and best practices. This step helps identify potential errors or mis-configurations.

[2] https://cloud.ibm.com/docs/secure-enterprise?topic=secure-enterprise-understanding-projects

6. Run Compliance Checks

   Before deployment, the project runs compliance checks to ensure that the configuration meets organizational, regulatory, and security requirements. Non-compliant configurations are flagged.

7. Approve

   Once compliance checks are passed, the configuration moves to the approval stage. Here, stakeholders or project owners provide final authorization for deployment.

8. Deploy

   After approval, the deployment process begins. IBM Cloud Schematics deploys the defined resources using Terraform scripts to automate the process.

9. Monitor

   Post-deployment, the resources are actively monitored for performance, health, and compliance, ensuring ongoing operational integrity.

10. Update

    Over time, configurations can be updated to address new requirements, fix issues, or enhance functionality. Updates follow the same process, ensuring continued compliance and alignment with project goals.

IBM Cloud IaC projects streamline resource provisioning and management through automation, reducing manual effort and errors while ensuring compliance with security standards via automated checks. Pre-configured architectures and reusable Terraform scripts accelerate deployment cycles, and the platform simplifies scalability to meet dynamic business demands. With centralized visibility and control, teams can monitor configurations, deployments, and resource health more effectively. This approach ensures consistent deployments across environments, minimizing configuration drift. Overall, this workflow follows best practices in DevSecOp, enhance agility, and maintain secure, compliant cloud environments.

1.4 Requirements

When using deployable architectures on IBM Power Virtual Server or IBM Cloud several requirements must be met to ensure a successful implementation. These requirements encompass account setup, network configurations, resource provisioning, and compliance considerations. Here are the key components:

IBM Cloud Account Requirements

A paid IBM Cloud account is necessary to create and manage Power Virtual Server instances. Users must register and verify their accounts before proceeding.

In addition, an IBM Cloud API key is required for programmatic access and management of cloud resources.

Power Virtual Server Service Instance Setup

After setting up the account, users must create a Power Systems Virtual Server service instance through the IBM Cloud dashboard. This involves selecting the appropriate resource group and providing a meaningful name for the instance.

VPC Landing Zone setup

The deployable architecture *Power Virtual Server with VPC landing zone* provides an automated deployment method to create an isolated Power Virtual Server workspace and connect it with IBM Cloud services and public internet.

Network Configuration

A private network is essential for secure communication between virtual machines (VMs) within the Power Virtual Server environment. Users need to create a private subnet associated with their service instance.

Depending on the architecture, organizations need to either set up Direct Link connections for dedicated bandwidth, or configure a site-to-site VPN for secure communication between on-premises environments and Power Virtual Server. IBM Cloud Transit Gateway is a service designed to simplify and enhance network management across IBM Cloud Virtual Private Cloud (VPC) networks within IBM Cloud. Network management components like DNS, NTP, Floating Public IP gateway, proxy servers and NFS as a Service might be installed as part of network management.

Image Requirements

For SAP HANA and VPC landing zone deployable architecture, supported operating systems include Red Hat Enterprise Linux (RHEL), AIX, IBM I and SUSE Linux Enterprise Server (SLES). Users can either use the images from the catalog or alternatively provide their own Linux images in Open Virtual Appliance (OVA) format.

For SAP installation, SAP installation binaries should be downloaded from IBM Cloud Object Storage to be used during the SAP installation.

Storage Requirements

Each Power Virtual Server instance requires a boot volume, users can choose the storage tier of the boot volume at provisioning. The boot volume is configured automatically during the deployment phase.

By addressing these requirements, organizations can effectively deploy architectures using IBM Power Virtual Server on IBM Cloud, ensuring that their infrastructure is secure, compliant, and optimized for performance.

1.4.1 Networking considerations

In deployable architectures using Power Virtual Server on IBM Cloud there are several network considerations that are essential to ensure efficient, secure and reliable connectivity. Here are the key aspects to focus on:

Connectivity Options

Depending on your requirements based on the organization’s needs, either public IP access (requiring internet access from the datacenter) or private IP access (using Direct Link) can be utilized. Each option has implications for security and performance. The following network options and configurations should be considered for connecting your Power Virtual Server instances to your enterprise resources – either on premises or in the cloud.

- Direct Link Connect

  This service provides dedicated, high-bandwidth connections between on-premises environments and Power Virtual Server instances. It is ideal for seamless data transfer and efficient application testing, particularly for organizations with significant data traffic needs.

- Site-to-Site VPN:

  Alternatively, a site-to-site VPN connection can be configured to securely connect an on-premises network to Power Virtual Server. This method allows for private IP communication, enhancing security while facilitating data exchange between environments.

- Transit Gateway

  IBM Cloud Transit Gateway is a service designed to simplify and enhance network management across IBM Cloud Virtual Private Cloud (VPC) networks within IBM Cloud. For more information on Transit Gateway see About IBM Cloud Transit Gateway.

- VPN Gateways

  For client-to-site VPN configurations, VPN Gateways are necessary. Organizations need to order these gateways in each Power Virtual Server location to enable communication between different Power Virtual Server instances. This setup may involve configuring GRE (Generic Routing Encapsulation) tunnels for internal communication and IPsec tunnels for secure external connections.

Redundancy and High Availability

As in any environment, planning for high availability and disaster recovery for your Power Virtual Server hosted resources is an important task. Consider these items:

- Network Redundancy

  Implementing redundant network paths can enhance reliability. This includes using multiple Direct Links or VPN connections to ensure continuous availability in case of a failure in one path.

- Disaster Recovery (DR)

  Establishing a remote DR site using Power Virtual Server can facilitate quick recovery of critical operations in the event of an outage at the primary location. Ensuring that network configurations support DR strategies is crucial for business continuity.

Performance Considerations

Planning for performance in the Power Virtual Server involves ensuring that the network connections provided are sufficient to support the expected workloads. Consider these items:

- Latency and Bandwidth

  Assessing latency and bandwidth requirements is vital, especially for applications that require real-time data processing or high throughput. Direct Link offers better performance compared to standard internet connections due to its dedicated nature.

- Traffic Management

  Using Quality of Service (QoS) settings can help prioritize critical application traffic over less important data flows, ensuring optimal performance under varying load conditions.

Security

Planning for security is important to protect the data in the Power Virtual Server instance. Consider the following:

- Encryption

  Ensure that all data transmitted over VPNs is encrypted to protect sensitive information from interception.

- Firewall Configuration

  Properly configure firewalls at both ends of the VPN connection to allow necessary traffic while blocking unauthorized access.

Monitoring and Management

To ensure continued success in your Power Virtual Server implementation the appropriate monitoring tools need to be installed, both performance and security compliance. Ensure that you have these important monitoring tools:

- Network Monitoring Tools

  Implementing monitoring solutions can help track network performance and detect issues proactively. Tools integrated with IBM Cloud services can provide insights into traffic patterns and potential bottlenecks.

- Compliance Monitoring

  Regularly review network configurations against compliance frameworks relevant to your industry, ensuring that all security measures meet necessary standards.

By addressing these network considerations, organizations can effectively deploy architectures using Power Virtual Server on IBM Cloud, ensuring secure, reliable, and high-performance connectivity tailored to their specific needs.

1.4.2 Benefits of Deployable Architectures

IBM Cloud deployable architectures offer pre-built, standardized patterns for rapidly deploying and managing applications and infrastructure. These blueprints accelerate development, boost efficiency, and strengthen security and compliance for complex cloud environments. Leveraging preconfigured components and best practices, they streamline deployments, minimizing manual effort. Integrated security best practices and adherence to industry regulations ensure a compliant and secure foundation. Optimized resource utilization within these predefined configurations helps organizations control cloud costs.

2

Deployable architectures

In Chapter 1, “Introduction to Deployable Architectures” on page 1 we introduced Deployable Architectures as implemented in IBM Cloud and IBM Power Virtual Server. As your cloud implementations become more complex, for example as you build multiple SAP HANA instances, having tools to help you quickly create those instances in a repeatable and trusted way becomes more important. This is what a deployable architecture is designed to do.

This chapter describes the different deployable architectures that are currently available on the IBM Cloud. We provide a list of deployable architectures that are already set up as patterns for your usage in building your cloud environments, whether in the IBM Cloud, in Power Virtual Server, or in some cases building a solution with components from both.

This chapter covers the following topics:

© Copyright IBM Corp. 2025.

2.1 Deployable architectures

Building complex cloud applications often involves managing multiple interconnected components. As the number of environments grows, manual configuration through graphical user interfaces (GUIs) becomes increasingly time-consuming and prone to errors.

Deployable architectures are automated scripts designed to streamline the process of building cloud environments. They offer significant benefits such as:

- Speed: Rapidly provision and configure environments.
- Consistency: Ensure repeatable deployments, reducing errors.
- Efficiency: Automate routine tasks, freeing up valuable time.

IBM Cloud provides a rich catalog of pre-built deployable architectures for common cloud tasks, helping clients accelerate their cloud adoption journey.

2.1.1 List of available deployable architectures

At the time of writing, 12 deployable architectures are available from the IBM Cloud Catalog, of which 2 are specifically for the IBM Power Virtual Server. Figure 2-1 shows these deployable architectures.

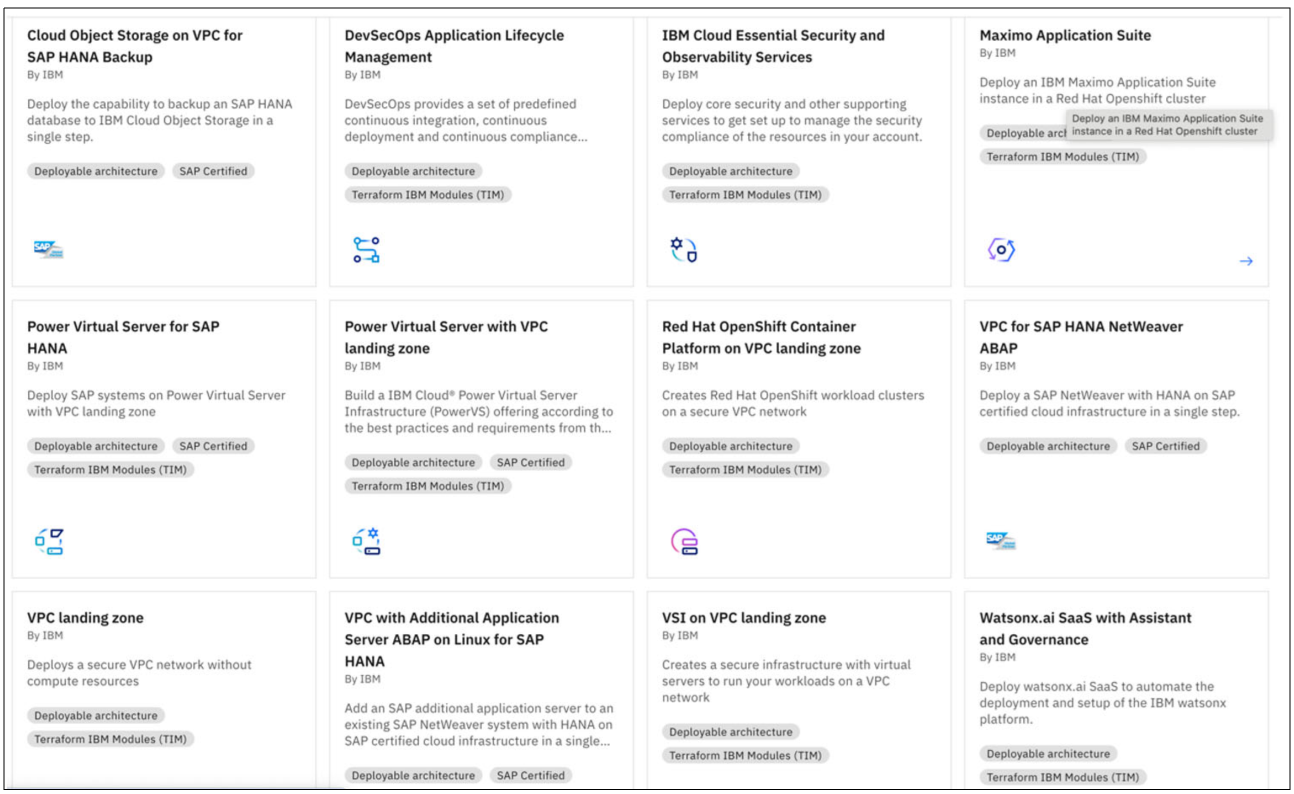


Figure 2-1 Overview of currently available deployable architectures from the IBM Cloud Catalog

Cloud Object Storage on VPC for SAP HANA Backup

This automation solution is designed for the implementation of an SAP HANA database backup solution using Backint and IBM Cloud Object Storage. The solution is based upon SAP note '2935898 - Install and Configure SAP HANA Backint Agent for Amazon S3'.

DevSecOps Application Lifecycle Management

The DevSecOps deployable architecture creates a set of DevOps Toolchains and pipelines. DevSecOps uses Continuous Delivery (Git Repositories and Issue Tracking, Tekton Pipelines, DevOps Insights, and Code Risk Analyzer), Secrets Manager, Key Protect, Cloud Object Storage, Container Registry and Vulnerability Advisor.

Out of the box, DevSecOps also leverages popular scanning tools such as SonarQube, GoSec, OWASP Zap (dynamic scan), any unit test framework, and GPG for image signing. It can also be used with more tools such as external Git providers and artifact stores. DevSecOps supports hybrid deployments, in particular by using private pipeline workers, and can be interfaced with other deployment tools such as Satellite Config.

IBM Cloud Essential Security and Observability Services

This deployable architecture is a preconfigured set of infrastructure as code (IaC) assets that are deployed and configured based on the recommended best practices. IBM Cloud's core security services are crucial for ensuring robust security and compliance for cloud-based applications and data. The primary goal is to provide a framework for secure and compliant IBM Cloud workloads.

The solution provideses a fully automated deployment of IBM Cloud core security services and its dependencies through IBM Cloud Project, providing a flexible and customizable foundation for your own application deployments on IBM Cloud.

By leveraging this architecture, deployments can be accelerated and tailored to meet unique business needs and enterprise goals.

This deployable architecture:

- Ensures observability by deploying services such as IBM Log Analysis, IBM Cloud Monitoring, IBM Cloud Activity Tracker Event Routing, IBM Cloud Event Notifications, and log retention through IBM Cloud Object Storage buckets.
- Implements and ensures security by deploying IBM Key Protect and IBM Cloud Secrets Manager.
- Achieves and ensures regulatory compliance by implementing centralized key management, centralized secrets management, and along with IBM Cloud Security and Compliance Center Workload Protection, secure application lifecycle management.

Maximo Application Suite

The IBM Maximo Application Suite deployable architecture provides a simple automated way to get started with Maximo Application Suite on IBM Cloud. It is a set of applications for asset monitoring, management, predictive maintenance, and reliability planning. The solution is a single, integrated cloud-based platform that uses Artificial Intelligence (AI), Internet of Things (IoT), and analytics to optimize performance, extend asset lifecycles, and reduce operational downtime and costs.

This deployable architecture automates the deployment of a standard client-managed Maximo Application Suite instance with its dependencies. It can also automatically enable the IBM Maximo Manage application. The suite allows to access configurable CMMS, EAM, APM, and RCM applications, along with streamlined installation and administration, and a unified user experience with shared data and workflows.

Power Virtual Server for SAP HANA

This is one of the two currently available deployable architectures for the Power Virtual Server. A full overview can be found in section 2.2.2, “Power Virtual Server for SAP HANA” on page 23.

Power Virtual Server with VPC landing zone

This is second one of the two currently available deployable architectures for the Power Virtual Server. A full overview can be found in section 2.2.1, “Power Virtual Server with VPC landing zone” on page 23.

Red Hat OpenShift Container Platform on VPC landing zone

This deployable architecture provides the tools to deploy a Red Hat OpenShift Container Platform cluster on IBM Cloud.

The deployable architecture deploys a Red Hat OpenShift cluster in a single VPC that is used to manage the platform in a VPC on IBM Cloud. The VPC is a multi-zoned, multi-subnet implementation that keeps your VPC secure and highly available.

Following features are supported:

- All features of the VPC landing zone deployable architecture.
- Creates a cluster in IBM Cloud VPC (version and cluster size can be specified).
- Creates worker pools in a container platform. These pools group and manage worker nodes with similar configurations (e.g. compute resources and availability zones.
- Configures subnets for the cluster, and specifies the subnets to deploy the worker nodes in.
- Configures private and public endpoints for the cluster.
- Configures the ingress controller for the cluster. This controller is responsible for routing external traffic to the appropriate services within the cluster.

VPC for SAP HANA NetWeaver ABAP

This automation solution is designed for the deployment of SAP NetWeaver on the HANA DB Stack. The SAP solution will be deployed on one of the following Operating Systems:

- SUSE Linux Enterprise Server 15 SP 3 for SAP
- SUSE Linux Enterprise Server 15 SP 4 for SAP
- Red Hat Enterprise Linux 8.4 for SAP
- Red Hat Enterprise Linux 8.6 for SAP

in an existing IBM Cloud Gen2 VPC, using an existing bastion host with secure remote SSH access.

VPC landing zone

The VPC landing zone deployable architectures are a preconfigured set of infrastructure as code (IaC) assets that are based on the IBM Cloud for Financial Services reference architecture.

Three deployable architectures are included: VPC landing zone, VSI on VPC landing zone, and Red Hat OpenShift Container Platform on VPC landing zone. This deployable architecture can be used to create a secure and customizable Virtual Private Cloud (VPC) environment.

You can use the VPC landing zone to deploy a simple IBM Cloud VPC infrastructure without any compute resources, such as Virtual Server Instances (VSIs) or Red Hat OpenShift clusters. You can also use it as a base on which to deploy your own compute resources.

The VPC landing zone is also a modular solution. You can use this architecture as the base for your compute resources. In fact, the other landing zone deployable architectures use the VPC landing zone as the base for their resources.

This deployable architecture supports following features:

- **Compliance**: Aligns with the VPC reference architecture for IBM Cloud for Financial Services.
- Creates a **VPC**-based topology based on two VPCs by default.
- Defines multiple **subnets** in the VPC to define IP ranges and organize resources within the network.
- Includes **public gateways** that provide connectivity between resources in a VPC and the public internet.
- Creates **access control lists (ACLs)** and define rules for allowing or denying traffic between subnets within a VPC.
- Creates a **transit gateway** to connect the two default VPCs that the deployable architecture creates.
- Creates **security groups** to control inbound and outbound traffic to resources within the VPC.
- Adds **key management** by integrating the IBM Key Protect for IBM Cloud service or the Hyper Protect Crypto Services. These key management services help to create, manage, and use encryption keys to protect your sensitive data.
- **Edge networking**: Isolates and speeds traffic to the public internet by using an edge VPC in a specific location, if enabled.
- Integrates **flow logs and activity tracking** services to enhance the observability and auditing of the VPC infrastructure.
- Adds landing zone VPC CRNs to an existing CBR (Context-based restrictions) network zone if the existing CBR zone ID is specified.

VPC with Additional Application Server ABAP on Linux for SAP HANA

This deployable architecture will add an SAP additional application server (SAP AAS NetWeaver 7.x on the SAP HANA-based ABAP stack) to an existing SAP NetWeaver system with HANA on SAP certified cloud infrastructure in a single step.

The Terraform scripts use the IBM Cloud Virtual Private Cloud (VPC) information that is provided and then call the Ansible playbook to create the SAP architecture on the specified VPC. IBM Cloud VPC infrastructure consists of SAP certified hardware that uses Intel CPU technologies.

VSI on VPC landing zone

The VSI on VPC landing zone deployable architecture deploys secure and compliant Virtual Server Instances (VSI) on top of an existing IBM Cloud Virtual Private Cloud (VPC) network. This deployable architecture provides secure and customizable compute resources for running applications and services.

This deployable architecture supports following features:

– All the features of the VPC landing zone deployable architecture
– **VSIs**: Creates and configures one or more virtual servers in an IBM Cloud VPC for workloads.
– Configures the **subnets for the VSIs**, and specifies which subnets the instances are deployed in.
– Associate **security groups** with the VSIs to control inbound and outbound traffic to instances.
– Provision and manage **SSH keys** for the VSIs so that instances can be securely administered.

The watsonx.ai SaaS with Assistant and Governance deployable architecture

This deployable architecture automates the deployment and setup of the IBM watsonx platform in an IBM Cloud account. The IBM watsonx platform is made of several services working together to offer AI capabilities to end users, who can explore them using IBM watsonx projects. The automation also configures a starter IBM watsonx project for an existing IBM Cloud user in the target IBM Cloud account.

The deployable architecture can also be used as part of a larger solution, where it is included in a stack of other components. This automation solution then provides output parameters that can be used programmatically for wiring the deployable architecture to the other components of the stack and has the flexibility to install additional Watson services.

The deployable architecture performs the following steps to create a ready to use IBM watsonx platform in a target IBM Cloud account, resource group, and IBM Cloud location:

– Creates or uses an existing resource group in the target IBM Cloud account.
– Creates the following services in the target resource group and location:
  - Cloud Object Storage
  - Watson Studio
  - Watson Machine Learning
– Optionally, it can create one or more of the following services in the target resource group and location:
  - watsonx.data
  - watsonx.governance
  - watsonx Assistant
  - Watson Discovery
  - watsonx IBM Orchestrate
– It creates the IBM watsonx user profile for an existing user in the target IBM Cloud account. This user is also referred as IBM watsonx admin.
– If you provided the CRN of a IBM Key Protect instance in the same target account and location of the watsonx services, then it enables storage delegation for the Cloud Object Storage instance.
– It creates a starter IBM watsonx project.

2.2 Deployable architectures for Power Virtual Server

Out of the twelve deployable architecture that are currently available, two are available for Power Virtual Server:

- Power Virtual Server with VPC landing zone
- Power Virtual Server for SAP HANA

2.2.1 Power Virtual Server with VPC landing zone

The Power Virtual Server with VPC landing zone is a deployable architecture offered by IBM Cloud which enables customers to create a secure and customizable Power Virtual Server environment with associated IBM Cloud resources. This architecture automates the deployment process, setting up an isolated Power Virtual Server workspace and connecting it to IBM Cloud services and the public internet. As part of this deployment, essential network management components, such as DNS, NTP, proxy servers, and NFS as a service, are automatically installed. By leveraging this offering, customers can significantly reduce deployment time, cutting down the process of setting up a Power Virtual Server workspace from several days to just one hour.

The Power Virtual Server with VPC landing zone deployable architecture is described in detail in Chapter 3, “Power Virtual Server with VPC landing zone” on page 25.

2.2.2 Power Virtual Server for SAP HANA

SAP solution provisioning on Power Virtual Server using deployable architectures provides an automated deployment method to create a Power Virtual Server landscape with SAP HANA systems.

When we compare this to the provisioning via the web user interface (webUI), user interaction is minimized, and SAP system deployment time is reduced from weeks to days.

SAP solution provisioning as a deployable architecture is two-step process and is composed of two Terraform based solutions:

- Power Virtual Server with VPC landing zone
- Power Virtual Server for SAP HANA

This deployable architecture is designed to assist you in deploying SAP ERP software landscapes into IBM Cloud on the IBM Power Virtual Server infrastructure. This is the second step in the deployment process for creating a full environment. Before starting this step, you should first deploy 'Power Virtual Server with VPC landing zone'. Once this is completed, you are prepared to start this step.

Power Virtual Server for SAP HANA creates and prepares Power Virtual Server instances for SAP HANA and SAP NetWeaver workloads. After deployment completes, you may (depending on the framework you chose) begin installing SAP on the configured instances or login to your newly created SAP instances directly.

The Power Virtual Server for SAP HANA deployable architecture is described in detail in Chapter 4, “Power Virtual Server for SAP HANA” on page 45.

3

Power Virtual Server with VPC landing zone

The Power Virtual Server with VPC landing zone deployable architecture provides a framework to build one or more Power Virtual Server instances using the best practices and requirements from the IBM Cloud. The deployable architecture provides an automated deployment method to create an isolated Power Virtual Server workspace and connect it with IBM Cloud services and public internet. Network management components like DNS, NTP, proxy servers and NFS as a Service might be installed.

Comparing the provisioning through the IBM Cloud user interface, user interaction is minimized, and ready-to-go deployment time of a Power Virtual Server workspace is reduced from days to less than 1 hour.

In this chapter we will provide a detailed overview of the Power Virtual Server with VPC landing zone deployable architecture, including an in-depth exploration of the architecture itself, various deployment options. We the planning and deployment process for each variation, key considerations during planning, as well as the input and output involved in the setup.

This chapter covers the following topics:

© Copyright IBM Corp. 2025.

3.1 Introduction

The Power Virtual Server with VPC Landing Zone deployable architecture offers a robust framework to establish one or more isolated Power Virtual Server instances within the IBM Cloud. This automated deployment method adheres to best practices and fulfills specific IBM Cloud requirements.

Key Benefits:

- Enhanced Security: Creates isolated Power Virtual Server workspaces, minimizing potential security risks.
- Simplified Management: Streamlines deployment and configuration processes through automation.
- Flexible Connectivity: Enables seamless integration with IBM Cloud services and public internet access.
- Comprehensive Network Management: Includes essential network components like DNS, NTP, proxy servers, and NFS as a Service for optimal performance and reliability.

By leveraging this architecture, organizations can confidently deploy and manage Power Virtual Server instances, ensuring a secure, efficient, and scalable solution.

The Power Virtual Server with VPC landing zone architecture makes use of the a terraform module created by IBM and published in github. If you want to get more information about this terraform module see powervs-vpc-landing-zone,

3.2 Variations of Power Virtual Server with VPC landing zone

The Power Virtual Server with VPC landing zone deployable architecture provides solutions that help provision VPC landing zones, Power Virtual Server workspaces, and interconnect them. The solutions are available in the IBM Cloud Catalog and can also be deployed without the catalog, except for the second solution below.

Four solutions are offered:

1. Standard

   Creates a VPC and Power Virtual Server workspace, interconnects them, and configures OS network management services (SQUID proxy, NTP, NFS, and DNS services) using Ansible Galaxy collection roles ibm.power_linux_sap collection.

2. Standard Extend

   Extends the standard solution by creating a new Power Virtual Server workspace in a different zone and interconnects with the previous solution.

   This solution is typically used for High Availability scenarios where a single management VPC can be used to reach both Power Virtual Server workspaces.

3. Quickstart (Standard plus VSI)

   Creates a VPC and a Power Virtual Server workspace, interconnects them, and configures operating network management services (SQUID proxy, NTP, NFS, and DNS services) using Ansible Galaxy collection roles ibm.power_linux_sap collection.

   Additionally creates a Power Virtual Server Instance of a selected t-shirt size.

   This solution is typically utilized for PoCs, demos, and quick onboarding to Power Virtual Server Infrastructure.

4. Import

   Takes information about an existing infrastructure and creates a schematics workspace.

   The schematics workspace's ID and the outputs from it can be used to install the terraform solution 'Power Virtual Server for SAP HANA' on top of a pre-existing Power Virtual Server infrastructure.

   It creates the ACL and security group rules necessary for management services (NTP, NFS, DNS, and proxy server) and schematics engine access.

   This solution is typically used for converting an existing Power Virtual Server landscape to a Schematics workspace.

Figure 3-1 shows the different variations and the actions that each varition completes.

Variation	Available on IBM Catalog	Requires IBM Schematics Workspace ID	Creates VPC Landing Zone	Performs VPC VSI OS Config	Creates PowerVS Infrastructure	Creates PowerVS Instance	Performs PowerVS OS Config
Standard	✓	N/A	✓	✓	✓	N/A	N/A
Standard Extend	✓	✓	N/A	N/A	✓	N/A	N/A
Quickstart (Standard plus VSI)	✓	N/A	✓	✓	✓	✓	N/A
Import	✓	N/A	N/A	N/A	N/A	N/A	N/A

Figure 3-1 Variations of Power Virtual Server with VPC landing zone

3.2.1 Standard variation

This variation of Power Virtual Server with VPC landing zone creates the VPC landing zone and the Power Virtual Server infrastructure, but does not create the Power Virtual Server instance. The following are created:

- A VPC Infrastructure with the following components:
  - One RHEL VSI for management (jump/bastion)
  - One RHEL VSI for network-services configured as squid proxy, NTP and DNS servers (using Ansible Galaxy collection roles ibm.power_linux_sap collection. This VSI also acts as central Ansible execution node
  - Optional Client to site VPN server
  - Optional File storage share
  - Optional Application load balancer
  - IBM Cloud Object storage (COS) Virtual Private endpoint gateway (VPE)
  - IBM Cloud Object storage (COS) Instance and buckets
  - VPC flow logs
  - KMS keys
  - Activity tracker
  - Optional Secrets Manager Instance with private certificate
  - A local or global transit gateway
- A Power Virtual Server workspace with the following network topology:
  - Creates two private networks: a management network and a backup network
  - Attaches Power Virtual Server workspace to transit gateway

– Creates an SSH key
– Optionally imports list of stock catalog images.
– Optionally imports up to three custom images from Cloud Object Storage.

The components created by this variation are shown in Figure 3-2.

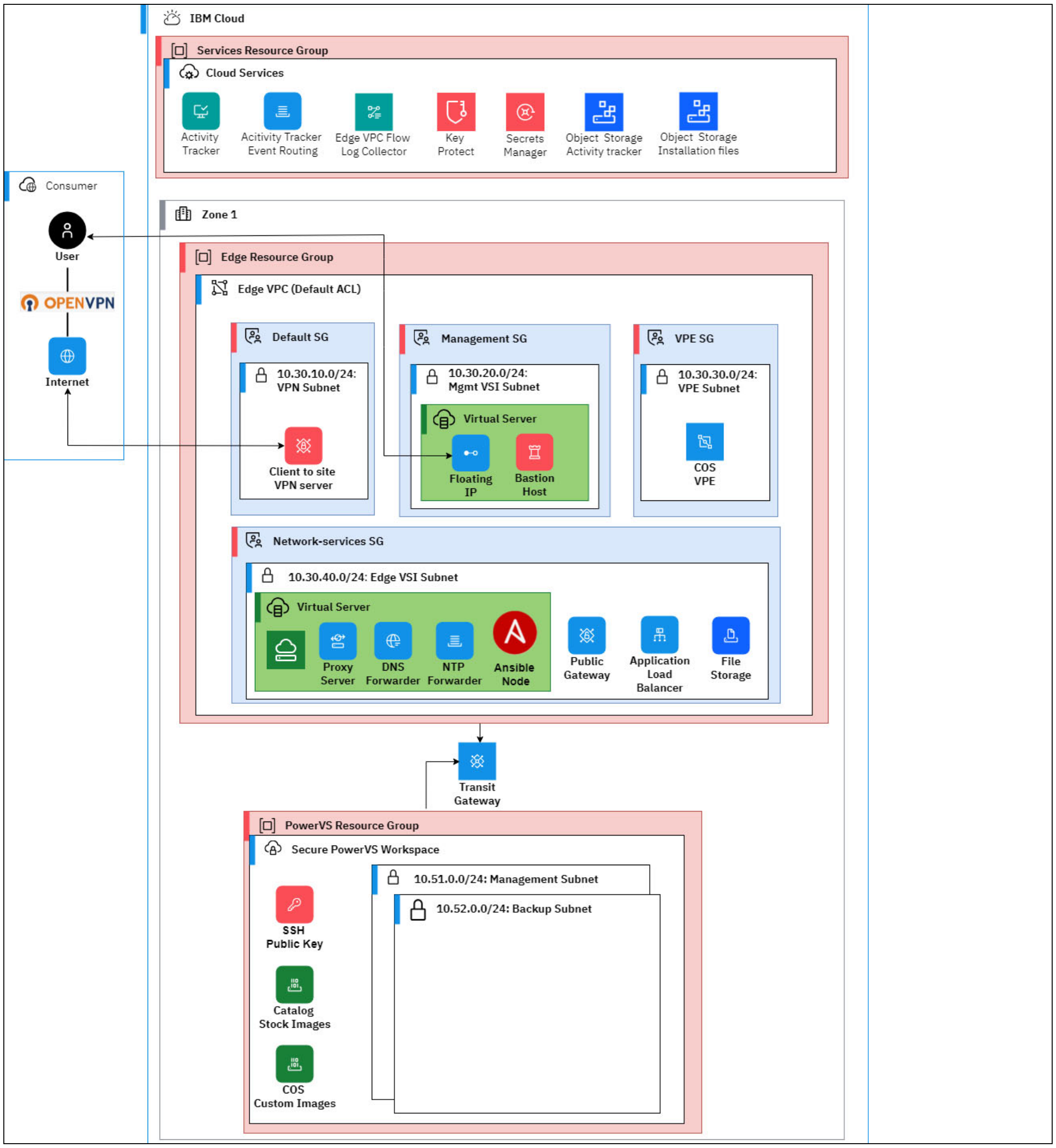


Figure 3-2 Architecture diagram for standard variation

3.2.2 Extend Power Virtual Server with VPC landing zone - Standard variation

The 'Extend Power Virtual Server with VPC landing zone' variation creates an additional Power Virtual Server workspace and connects it to the existing Power Virtual Server with VPC landing zone. It builds on existing Power Virtual Server with VPC landing zone deployed as a variation 'Create a new architecture'. This is typically used for High Availability scenarios in the same regions.

Important: This variation has a prerequisite. You must deploy the 'Create a new architecture Standard' variant first.

This deployable architecture variation supports these features:

- A Power Virtual Server workspace with the following network topology:
  - Creates two private networks: a management network and a backup network.
  - Attaches the Power Virtual Server workspace to transit gateway.
  - Creates an SSH key.
  - Optionally imports list of stock catalog images.
  - Optionally imports up to three custom images from Cloud Object Storage.

The components created by this variation are shown in Figure 3-3.

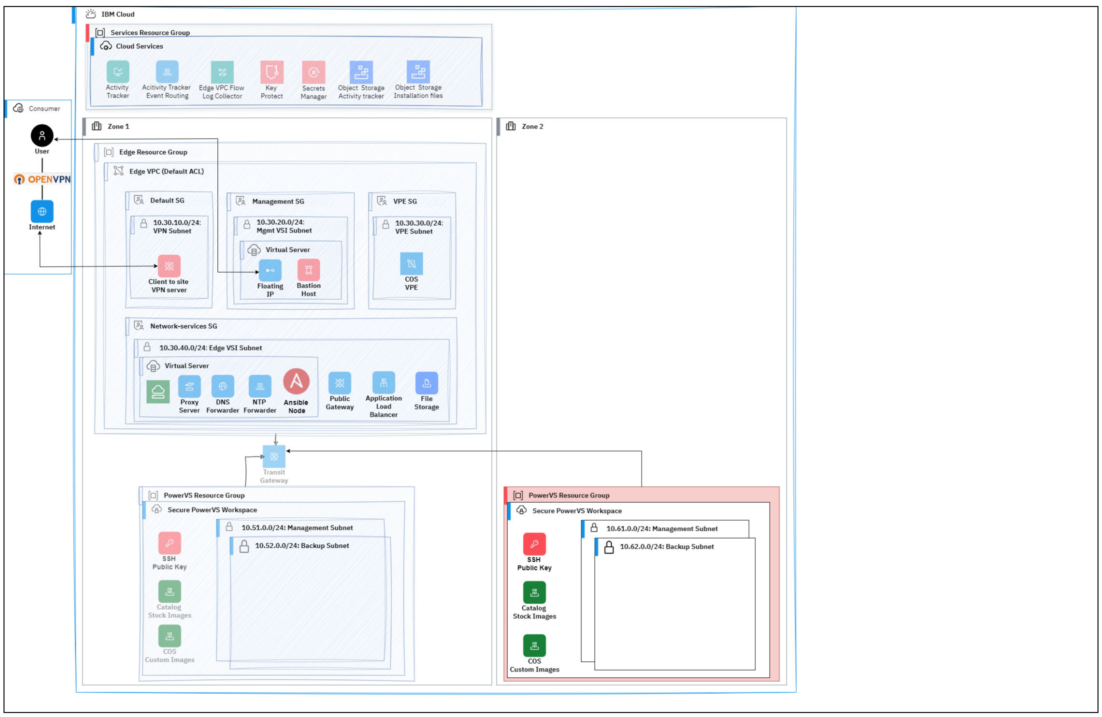


Figure 3-3 Architecture diagram for extend variation

3.2.3 Quickstart variation

This deployable architecture variation supports these features:

- A VPC Infrastructure with the following components:
  - One RHEL VSI for management (jump/bastion)
  - One RHEL VSI for network-services configured as squid proxy, NTP and DNS servers (using Ansible Galaxy collection roles ibm.power_linux_sap collection. This VSI also acts as central Ansible execution node
  - Optional Client to site VPN server
  - Optional File storage share
  - Optional Application load balancer
  - IBM Cloud Object storage (COS) Virtual Private endpoint gateway (VPE)
  - IBM Cloud Object storage (COS) Instance and buckets
  - VPC flow logs

– KMS keys
– Activity tracker
– Optional Secrets Manager Instance with private certificate
– A local or global transit gateway

- A Power Virtual Server workspace with the following network topology:
  – Creates two private networks: a management network and a backup network.
  – Attaches the Power Virtual Server workspace to transit gateway.
  – Creates an SSH key.
  – Imports cloud catalog stock images.
  – A Power Virtual Server Instance with following options:
    • t-shirt profile (AIX/IBM i/SAP Image)
    • Custom profile (cores, memory, storage and image)
    • 1 volume

You can run AIX, IBM i, and Linux images on your virtual server instances. Select the required T-shirt size and a virtual server instance with the appropriate T-shirt size is deployed. Custom configuration is also supported. The components created by this variation are shown in Figure 3-4.

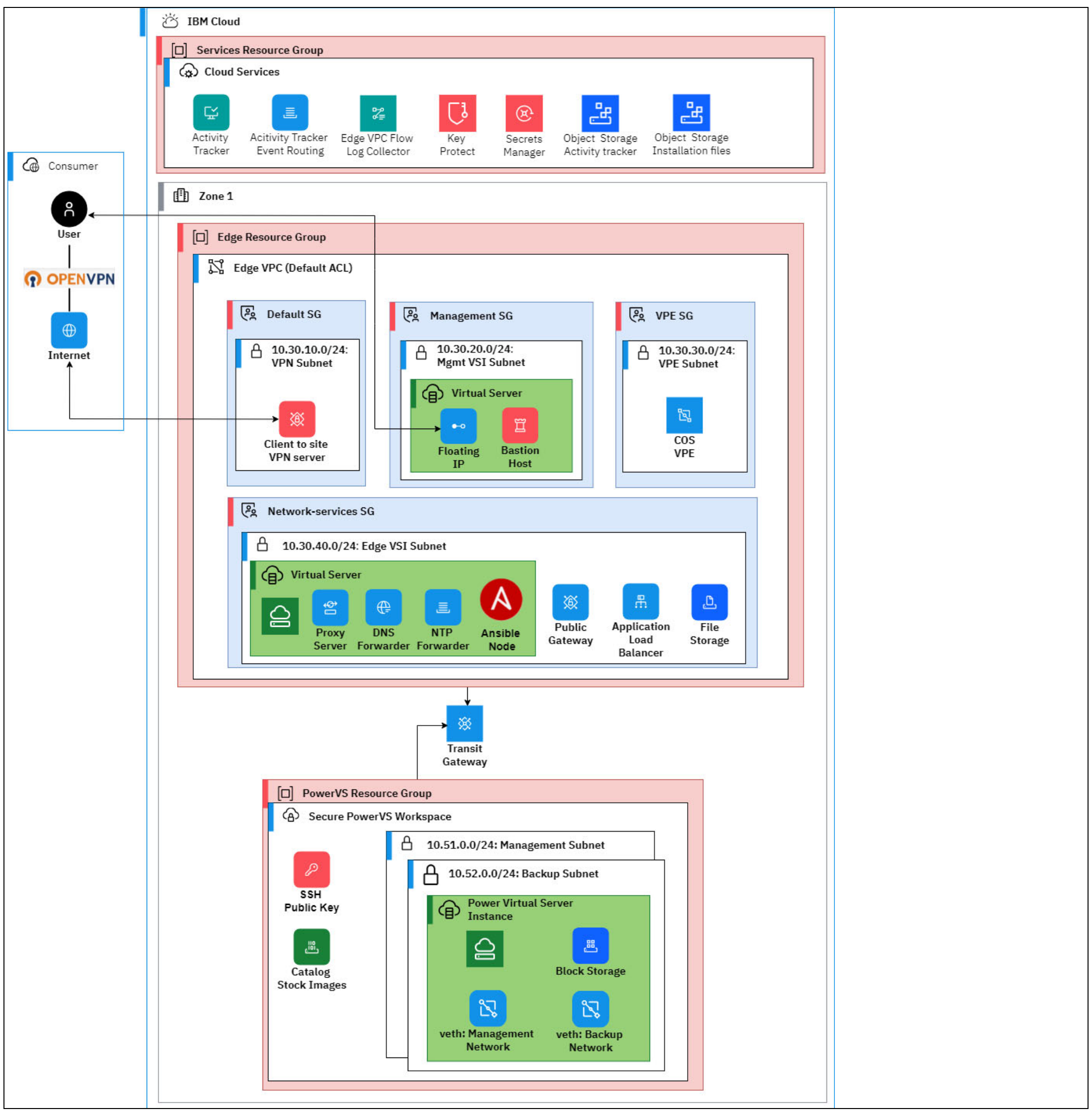


Figure 3-4 Architecture diagram for extend variation

3.2.4 Import Variation

This solution helps to install the deployable architecture *Power Virtual Server for SAP HANA* on top of a pre-existing Power Virtual Server landscape. *Power Virtual Server for SAP HANA* automation requires a schematics workspace id for installation. The 'Import' solution creates a schematics workspace by taking pre-existing VPC and Power Virtual Server infrastructure resource details as inputs. The ID of this schematics workspace will be the pre-requisite workspace id required by 'Power Virtual Server for SAP HANA' to create and configure the Power Virtual Server instances for SAP on top of the existing infrastructure.

Prerequisites

The preexisting infrastructure must meet the following conditions to use the 'Import' solution to create a schematics workspace:

- Virtual Private Cloud (VPC) side
  - Existing VPC or VPCs with virtual server instances, ACL/ACLs, and Security Groups.
  - Existing access host (jump server) which is an intel based virtual server instance that can access Power virtual server instances.
  - Existing Transit Gateway.
  - The VPC in which the jump host exists must be attached to the Transit Gateway.
  - The necessary ACLs and security group rules for VPC in which the access host (jump server) exists must allow SSH login to the Power virtual server instances which would be created using 'Power Virtual Server for SAP HANA' automation.
- Power Virtual Server Workspace side
  - Existing Power Virtual Server Workspace with at-least two private subnets.
  - Power Virtual Server Workspace/Cloud Connections must be attached to above Transit Gateway.
  - SSH key pairs used to login to access host/jump host (Intel based virtual server instance) on VPC side should match to the existing SSH key used in Power Virtual Server Workspace.
- Mandatory Management Network Services
  - Existing Proxy server ip and port required to configure the internet access required for Power Virtual Server instances.
- Optional Management Network Services
  - Existing DNS server ip for the Power Virtual Server instances.
  - Existing NTP server ip for the Power Virtual Server instances.
  - Existing NFS server ip and path for the Power Virtual Server instances.

If the above parameters are provided, then it must be made sure IPs are reachable on Power virtual server instances which would be created using 'Power Virtual Server for SAP HANA' automation.

Note: IBM Cloud has a quota of 100 ACL rules per ACL. The 'Import' variation will create 52 new ACL rules for providing schematics servers access to the access host (this access is required for 'Power Virtual Server for SAP HANA' automation). Please ensure the concerned ACL can take in new ACL rules without exceeding the quota of 100 so the deployment will be successful.

Resources Created:

ACL rules for IBM Cloud Schematics are created for the VPC subnets in which access host(jump server) exists.

Schematics workspace required by 'Power Virtual Server for SAP HANA' to create and configure the Power Virtual Server instances for SAP on top of the existing infrastructure.

The components created by this variation are shown in Figure 3-5.

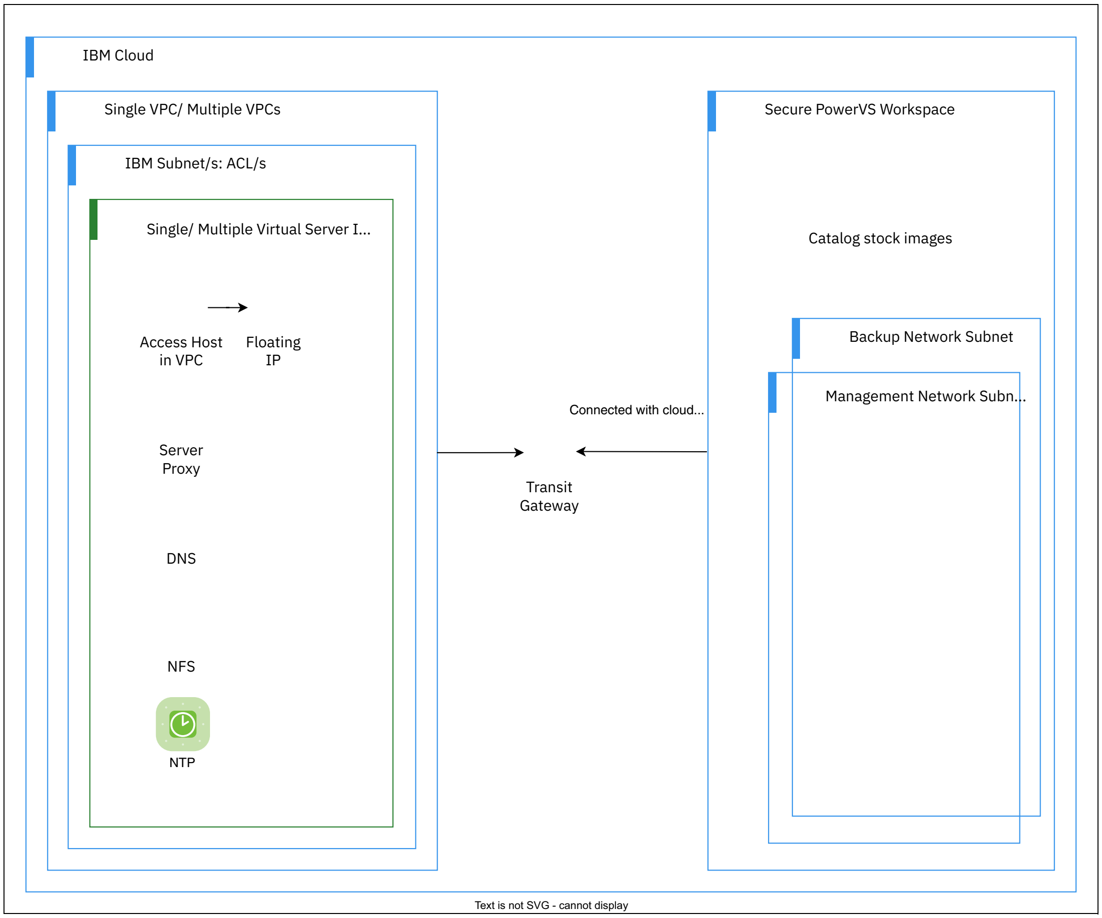


Figure 3-5 Architecture diagram for import variation

3.3 Considerations and planning

As part of security, ACL (access control list) rules are set as shown in Table 3-1.

Table 3-1 ACL rule settings

Rule type	Priority	Allow or Deny	Protocol	Source	Destination
Inbound	1	Allow	ALL	ANY IP	ANY IP
Outbound	1	Allow	ALL	ANY IP	ANY IP

Security groups are created and attached to correct subnets/VPE/VPN. For the management security group, 25 schematics IP addresses are added to inbound rules. This is required to allow ssh login access from schematics to the intel VSI (Virtual Server Instances) to perform OS configuration using ansible playbooks.

Table 3-2 on page 33 shows the security group rules.

Table 3-2 Security group rules

Name	Source	Protocol: Value	Attached resources
management-sg	Schematic IP addresses IBM inbound 161.26.0.0/16 10.0.0.0/8 172.16.0.0/12, 192.168.0.0/16 Optional user provided IP address/CIDR	TCP: 22 ALL : - TCP : 22 TCP : 22	prefix-jum-box-001 VSI
network-services-sg	IBM inbound 161.26.0.0/16 10.0.0.0/8 172.16.0.0/12, 192.168.0.0/16	ALL ALL	prefix-network-services-001-VSI, load balancer, mount share targets
vpe-sg	IBM inbound 161.26.0.0/16 10.0.0.0/8 172.16.0.0/12, 192.168.0.0/16	ALL ALL	Cloud Object Storage

For RHEL VSIs below are the private networks that are created by the deployment automation with corresponding default values

Subnet name	Private Network	IP Address ranges
prefix-edge-vpn-zone-1	Private Network for VPN Server	10.30.10.0/24
prefix-edge-vsi-management-zone-1	Management network for Bastion virtual server instance	10.30.20.0/24
prefix-edge-vpe-zone-1	Private network for Cloud Object storage VPE	10.30.30.0/24
prefix-edge-vsi-edge-zone-1	Private network for Network services VSI. This subnet has public gateway enabled.	10.30.40.0/24

3.4 Inputs and outputs

There are multiple modules used in the Power Virtual Server with VPC landing zone deployable architecture. These modules are linked by input and output variables to allow the automated build of the infrastructure.

3.4.1 Inputs

The success of deploying a VPC landing zone with IBM Power Virtual Server relies heavily on preparing the right input configurations. Below are the detailed technical aspects to consider:

Network Classless Inter-Domain Routings and Security Group Configurations

The use the following guidelines for defining your Classless Inter-Domain Routing (CIDR) and security groups.

1. CIDR Block Design:

   Define the IP address ranges for the VPC and subnets using Classless Inter-Domain Routing (CIDR).

   For example, use 10.0.0.0/16 for the VPC and divide it into smaller subnets (e.g., 10.0.1.0/24 for application servers, 10.0.2.0/24 for databases).

   Ensure non-overlapping IP ranges to avoid conflicts in hybrid cloud environments.

2. Subnet Design

   When defining your subnets, choose the appropriate subnet type for each network requirement.

   – Use private subnets for secure internal workloads, such as databases or internal APIs.
   – Use public subnets for workloads that need external access, such as web servers.

3. Security Groups:

   Define inbound and outbound rules for traffic control. For example Allow SSH (port 22) only from known IP addresses for administrative access.

   Restrict database access (port 3306) to only specific application servers.

4. Firewall Rules:

   Implement or stateless firewall rules for further packet filtering.

   Enable default deny all rules and whitelist necessary traffic flows.

Workload Sizing

The following are guidelines for sizing the Power Virtual Server environment to run your specific workloads.

1. vCPU, Memory, and Storage:

   Sizing depends on the workload type:

   – General Compute

     For moderate workloads, start with 2-4 vCPUs and 16-32 GB RAM. For Memory-Intensive Workloads (e.g., SAP HANA), opt for configurations like 16 vCPUs and 256 GB RAM.

   – Storage

     Boot Disk should be minimum 50 GB.

     Data Volumes: should be defined based on application needs. Choose the appropriate IOPS level for your data volumes.

   Use IBM's Power Sizer Tool to determine the optimal configuration.

2. High Availability (HA) Considerations:

   Ensure redundant instances across availability zones. Leverage load balancers for application scaling.

Identity and Access Management Roles, Access Policies and key management

Consider the following when defining your Identity and Access Management (IAM) roles:

1. Use least privilege principles to minimize risk.
2. Create specific IAM roles for:
   - Administrators: Full access to manage resources.
   - Developers: Access to development environments only.
   - Auditors: Read-only access for compliance monitoring.

   When defining your Access Policies define granular policies using JSON as shown in Example 3-1.

   Example 3-1 Security policy definition to limit an IAM role to access specific VPC resources only

```
: "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": "ec2:TerminateInstances",
      "Resource": "*"
    }
  ]
```

3. For key management use IBM Key Protect for managing encryption keys for storage and network security.

Steps for Setting Up Terraform Templates

The deployable architecture uses Terraform for defining the infrastructure. The following steps are required to be configured:

1. Define Terraform Configuration:

   Create a Terraform configuration file (main.tf) to define your VPC, subnets, and instances. as shown in Example 3-2.

   Example 3-2 Terraform configuration file example

```
provider "ibm" {
  region = "us-south"
resource "ibm_is_vpc" "example" {
  name = "example-vpc"
  address_prefix_management = "manual"
resource "ibm_is_subnet" "public" {
  name = "public-subnet"
  ip_version = "ipv4"
  vpc = ibm_is_vpc.example.id
  total_ipv4_address_count = 256
  zone = "us-south-1"
```

2. Deploy and Validate

   The following steps are used to deploy and validate the Terraform actions.

   a. Run terraform init, terraform plan, and terraform apply to provision resources.

   b. Validate the deployment by checking the VPC and subnet status in the IBM Cloud console.

3. Version Control

   Use Git repositories for managing and versioning Terraform templates.

Example Configuration

Example 3-3 shows a sample YAML Configuration to define input parameters for repeatable deployments.

Example 3-3 Sample YAML file

```
vpc:
  name: example-vpc
  region: us-south
  subnets:
   - name: public-subnet
      cidr: 10.0.1.0/24
      zone: us-south-1
```

Table 3-3 shows the input variables used in the deployable architecture.

Table 3-3 Inputs used within the modules for the Power Virtual Server with VPC landing zone deployable architecture

Name	Description	Type	Default	Required
IC_SCHEMATICS_WORKSPACE_ID	leave blank if running locally. This variable will be automatically populated if running from an IBM Cloud Schematics workspace	string	""	no
certificate_template_name	The name of the Certificate Template to create for a private_cert secret engine. When var.existing_sm_instance_guid is not null, then it has to be the existing template name that exists in the private cert engine.	string	"my-template"	no

Name	Description	Type	Default	Required
client_to_site_vpn	VPN configuration - the client ip pool and list of users email ids to access the environment. If enabled, then a Secret Manager instance is also provisioned with certificates generated. See optional parameters to reuse existing certificate from secrets manager instance.	object({ enable = bool client_ip_pool = string vpn_client_access_group_users = list(string) })	{ "client_ip_pool": "192.168.0.0/16", "enable": false, "vpn_client_access_group_users": [] }	no
configure_dns_forwarder	Specify if DNS forwarder will be configured. This will allow you to use central DNS servers (e.g. IBM Cloud DNS servers) sitting outside of the created IBM Power Virtual Server infrastructure. If yes, ensure 'dns_forwarder_config' optional variable is set properly. DNS forwarder will be installed on the network-services vsi.	bool	true	no
configure_nfs_server	Specify if NFS server will be configured. This will allow you easily to share files between Power Virtual Server instances (e.g., SAP installation files). File storage share and mount target in VPC will be created.. If yes, ensure 'nfs_server_config' optional variable is set properly below. Default value is '200GB' which will be mounted on specified directory in network-service vsi.	bool	true	no
configure_ntp_forwarder	Specify if NTP forwarder will be configured. This will allow you to synchronize time between IBM Power Virtual Server instances. NTP forwarder will be installed on the network-services vsi.	bool	true	no

Name	Description	Type	Default	Required

custom_profile	Overrides t-shirt profile: Custom Power Virtual Server instance. Specify 'sap_profile_id' here or combination of 'cores' & 'memory'. Optionally volumes can be created.	object({
		sap_profile_id = string
		cores = string
		memory = string
		server_type = string
		proc_type = string
		storage = object({
		size = string
		tier = string
		})
		})
			{
			"cores": "",
			"memory": "",
			"proc_type": "",
			"sap_profile_id": null,
			"server_type": "",
			"storage": {
			"size": "",
			"tier": ""
			}
			}
				no

custom_profile_instance_boot_image	Override the t-shirt size specs of Power Virtual Server Workspace instance by selecting an image name and providing valid 'custom_profile' optional parameter.	string	"none"	no

dns_forwarder_config	Configuration for the DNS forwarder to a DNS service that is not reachable directly from Power Virtual Server.	object({
		dns_servers = string
		})
			{
			"dns_servers": "161.26.0.7; 161.26.0.8; 9.9.9.9;"
			}
				no

existing_sm_instance_guid	An existing Secrets Manager GUID. The existing Secret Manager instance must have private certificate engine configured. If not provided an new instance will be provisioned.	string	null	no
existing_sm_instance_region	Required if value is passed into var.existing_sm_instance_guid.	string	null	no
external_access_ip	Specify the source IP address or CIDR for login through SSH to the environment after deployment. Access to the environment will be allowed only from this IP address. Can be set to 'null' if you choose to use client to site vpn.	string	n/a	yes

Name	Description	Type	Default	Required
ibmcloud_api_key	The IBM Cloud platform API key needed to deploy IAM enabled resources.	string	n/a	yes
network_services_vsi_profile	Compute profile configuration of the network services vsi (cpu and memory configuration). Must be one of the supported profiles. See here	string	"cx2-2x4"	no
nfs_server_config	Configuration for the NFS server. 'size' is in GB, 'iops' is maximum input/output operation performance bandwidth per second, 'mount_path' defines the target mount point on os. Set 'configure_nfs_server' to false to ignore creating file storage share.	object({ size = number iops = number mount_path = string })	{ "iops": 600, "mount_path": "/nfs", "size": 200 }	no
powervs_backup_network	Name of the IBM Cloud Power Virtual Server backup network and CIDR to create.	object({ name = string cidr = string })	{ "cidr": "10.52.0.0/24", "name": "bkp_net" }	no
powervs_management_network	Name of the IBM Cloud Power Virtual Server management subnet and CIDR to create.	object({ name = string cidr = string })	{ "cidr": "10.51.0.0/24", "name": "mgmt_net" }	no
powervs_resource_group_name	Existing IBM Cloud resource group name.	string	"Default"	no
powervs_zone	IBM Cloud data center location where IBM Power Virtual Server infrastructure will be created.	string	n/a	yes
prefix	A unique identifier for resources. Must begin with a lowercase letter and end with a lowercase letter or number. This prefix will be prepended to any resources provisioned by this template. Prefixes must be 16 or fewer characters.	string	n/a	yes

Name	Description	Type	Default	Required
sm_service_plan	The service/pricing plan to use when provisioning a new Secrets Manager instance. Allowed values: standard and trial. Only used if existing_sm_instance_guid is set to null.	string	"standard"	no
ssh_private_key	Private SSH key (RSA format) to login to Intel VSIs to configure network management services (SQUID, NTP, DNS and ansible). Should match to public SSH key referenced by 'ssh_public_key'. The key is not uploaded or stored. For more information about SSH keys, see SSH keys	string	n/a	yes
ssh_public_key	Public SSH Key for VSI creation. Must be an RSA key with a key size of either 2048 bits or 4096 bits (recommended). Must be a valid SSH key that does not already exist in the deployment region.	string	n/a	yes
tags	List of tag names for the IBM Cloud Power Virtual Server workspace	list(string)	[]	no
tshirt_size	Power Virtual Server instance profiles. These profiles can be overridden by specifying 'custom_profile_instance_boot_image' and 'custom_profile' values in optional parameters.	object({ tshirt_size = string image = string })	n/a	yes

3.4.2 Outputs

After deploying the VPC landing zone, you can expect the following outputs:

Configured VPC with Subnets and Routing Tables

1. VPC:
   - A fully functional VPC with defined subnets, each assigned to an availability zone.
   - Properly segmented public and private subnets.

2. Routing Tables:
   - Default route for outbound traffic to the internet via an Internet Gateway.
   - Private subnets route traffic through a NAT Gateway for secure outbound communication.

Deployment-Ready Power Virtual Server Instances

1. Power Virtual Server Instances:
   - Compute instances with pre-configured vCPU, memory, and storage, ready for application deployment.
   - Networking setup to ensure connectivity with VPC resources.
2. Access Control:
   - IAM roles and security group rules applied to all instances for controlled access.

Monitoring and Logging Setups

1. IBM Cloud Monitoring:
   - Enable monitoring to track VPC performance, instance health, and network latency.
   - Use IBM Cloud Log Analysis to centralize and visualize logs from all resources.
2. Tools:
   - Integrate with Prometheus or Grafana for advanced monitoring.
   - Enable alerts for specific metrics (e.g., CPU usage, network throughput).

3.4.3 Checklist for Verifying Deployment

Validate the following components:

- VPC CIDR blocks and subnet assignments.
- Security group rules and firewall policies.
- Route table configurations and internet connectivity.
- IAM policies and access control compliance.

3.4.4 Recommended Tools

Consider utilizing the following tools.

1. IBM Cloud Satellite:

   Use Satellite to manage hybrid environments. This provides Centralized visibility of on-premises and cloud VPC resources.
2. Terraform State Management:

   Track resource configurations and changes to ensure consistent deployments.

3.4.5 Output Variables

Table 3-4 on page 42 shows the outputs created by the Power Virtual Server with VPC landing zone deployable architecture.

Table 3-4   Outputs used within the modules for the Power Virtual Server with VPC landing zone deployable arehitecture

Name	Description
access_host_or_ip	Access host(jump/bastion) for created Power Virtual Server infrastructure.
ansible_host_or_ip	Central Ansible node private IP address.
dns_host_or_ip	DNS forwarder host for created Power Virtual Server infrastructure.
network_services_config	Complete configuration of network management services.
nfs_host_or_ip_path	NFS host for created Power Virtual Server infrastructure.
ntp_host_or_ip	NTP host for created Power Virtual Server infrastructure.
powervs_backup_subnet	Name, ID and CIDR of backup private network in created Power Virtual Server infrastructure.
powervs_images	Object containing imported Power Virtual Server image names and image ids.
powervs_instance_management_ip	IP address of the primary network interface of IBM Power Virtual Server instance.
powervs_instance_private_ips	All private IP addresses (as a list) of IBM Power Virtual Server instance.
powervs_management_subnet	Name, ID and CIDR of management private network in created Power Virtual Server infrastructure.
powervs_resource_group_name	IBM Cloud resource group where Power Virtual Server infrastructure is created.
powervs_ssh_public_key	SSH public key name and value in created Power Virtual Server infrastructure.
powervs_storage_configuration	Storage configuration of Power Virtual Server instance.
powervs_workspace_guid	Power Virtual Server infrastructure workspace guid. The GUID of the resource instance.
powervs_workspace_id	Power Virtual Server infrastructure workspace id. The unique identifier of the new resource instance.
powervs_workspace_name	Power Virtual Server infrastructure workspace name.
powervs_zone	Zone where Power Virtual Server infrastructure is created.
prefix	The prefix that is associated with all resources
proxy_host_or_ip_port	Proxy host:port for created Power Virtual Server infrastructure.
resource_group_data	List of resource groups data used within landing zone.
schematics_workspace_id	ID of the IBM Cloud Schematics workspace. Returns null if not ran in Schematics.
ssh_public_key	The string value of the ssh public key used when deploying VPC
transit_gateway_id	The ID of transit gateway.
transit_gateway_name	The name of the transit gateway.
vpc_names	A list of the names of the VPC.
vsi_list	A list of VSI with name, id, zone, and primary ipv4 address, VPC Name, and floating IP.

Name	Description
vsi_names	A list of the vsis names provisioned within the VPCs.

4

Power Virtual Server for SAP HANA

The IBM Cloud Power Virtual Server for SAP HANA deployable architecture is a Terraform-based solution designed to automate the deployment and configuration of Power Virtual Server instances specifically for SAP HANA and SAP NetWeaver workloads. It builds upon the foundation of the Power Virtual Server with VPC landing zone deployable architecture, which provides the underlying network infrastructure and security settings.

This chapter describes the Power Virtual Server for SAP HANA deployable architecture.

The following topics are covered:

© Copyright IBM Corp. 2025.

4.1 Introduction

The Power Virtual Server for SAP HANA deployable architecture deploys and configures Power Virtual Server instances optimized for SAP HANA and SAP NetWeaver, including instances for the HANA database, application servers, and shared file systems and optionally installs SAP S/4HANA or BW/4HANA software on the deployed instances, streamlining the deployment process. The deployable architecture leverages the network infrastructure provided by the VPC landing zone, including DNS, NTP, and NFS services and implements security best practices to protect the SAP environment, such as network segmentation, access controls, and encryption.

Traditional SAP solution provisioning on Power Virtual Server often involves manual steps and can be a time-consuming process. However, by leveraging deployable architectures, organizations can significantly streamline and expedite this process.

The key benefits of deployable architectures are

- Reduced Deployment Time: By automating many manual tasks, deployment time can be reduced from weeks to days.
- Minimized User Interaction: The need for manual intervention is significantly reduced, leading to fewer errors and increased efficiency.
- Enhanced Consistency: Deployable architectures ensure consistent and repeatable deployments, adhering to best practices.
- Improved Security: Automated deployments can help mitigate security risks by following standardized procedure.

SAP solution provisioning through the Power Virtual Server for SAP HANA deployable architectures is a composition of two Terraform based solutions:

1. IBM Power Virtual Server with VPC landing zone
2. IBM Power Virtual Server for SAP HANA.

Overview of SAP HANA on IBM Power Virtual Server

SAP HANA is a high-performance in-memory database optimized for real-time analytics and processing. IBM Power Virtual Server supports SAP HANA workloads with superior compute and memory performance, leveraging Power10 processors which are known for high throughput and efficient parallel processing.

Power10 systems provide large memory configurations that align with SAP HANA's in-memory architecture requirements. They also provide built-in virtualization using IBM PowerVM which ensures optimal resource utilization and workload isolation. Power Virtual Server supports SAP HANA-certified operating systems, including Red Hat Enterprise Linux (RHEL) for SAP Solutions and SUSE Linux Enterprise Server (SLES) for SAP Applications.

IBM Power Virtual Server provides high-performance SAN attached storage to meet the SAP HANA requirements for ultra-low latency and high IOPS to support SAP HANA data and log volumes.

SAP HANA is an advanced in-memory database designed for real-time analytics and processing. SAP HANA running on IBM Power benefits from the enhanced performance and memory bandwidth of the IBM Power processors. These performance benefits are available on IBM Power Virtual Server providing optimized performance for SAP workloads. Power Virtual Server supports SAP-certified configurations, providing high availability, scalability, and the flexibility of cloud or hybrid environments.

Some of the benefits of running SAP HANA on IBM Power Virtual servers are:

- Analytics Performance

  IBM Power is optimized for high-volume data analytics and complex queries, delivering low-latency insights. With on-chip AI support, Power is ideal for machine learning and advanced analytics workloads.

- Database Efficiency

  The in-memory architecture allows near-instantaneous data access and processing. This is enhanced as IBM Power10 processors provide high memory bandwidth and low latency memory access along with industry leading processing power.

- Reliability

  Power Virtual Server is designed with redundant components for resiliency and supports disaster recovery features important for SAP workloads.

- Compatibility:

  IBM Power Virtual server is fully compatible with SAP-certified operating systems, both Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES). The infrastructure is tailored for SAP HANA workloads, ensuring seamless integration with existing IT ecosystems.

Single-instance deployment

A single-instance deployment option is ideal for smaller workloads or test environments. This architectural patter provides a single SAP HANA database instance deployed on a dedicated Power Virtual Server instance. This option provides simple setup, lower costs, and faster deployment time.

Multi-Instance Deployment

A multi-instance deployment is designed for enterprise-grade workloads requiring high availability and disaster recovery. This pattern produces multiple SAP HANA instances distributed across Power Virtual Server instances for load balancing and failover capabilities. Multiple tenants are supported using SAP HANA Multitenant Database Containers (MDC).

When combined with SAP HANA System Replication which enables real-time replication between primary and secondary instance, the multi-instance deployment should be deployed across separate IBM Cloud regions to ensure geographic redundancy. For automated failover and enhanced availability leverage IBM PowerHA for SAP HANA.Benefits of running SAP HANA on IBM Power Virtual Server

Client Success Stories

A manufacturing company in India discovered in an internal review that their business-critical SAP HANA workload was hosted on-premises and was not compliant with disaster recovery standards. They evaluated several options and chose to run their SAP workload on Power Virtual Server as it provided the performance and reliability that met the client's business requirements. Migrating their SAP workload to IBM Power Virtual Server provided:

- Enhanced business continuity planning with reliable failover solutions, including disaster recovery.
- Cost savings by shifting from CapEx to Opex spending.

Another client, a large brewery with significant exports around the world, was concerned that their production workloads and large databases were on-premises and with political conflicts in Europe escalating, there was increasing concerns about security.

In addition, they were seeking a cloud solution that would allow them to easily expand their infrastructure. As a result of moving both their production and nonproduction workloads to Power Virtual Server:

- Client processing performance increased approximately 30%.
- The Cloud Platform solution reduced the CapEx costs and allowed budget for further improvements.
- Cloud removed capacity restriction and eliminated worries of having to buy new hardware.
- Allowed the use of the existing personnel to manage the remaining legacy systems and the IBM Cloud for SAP.

4.2 Variations

The Power Virtual Server for SAP HANA deployable architecture builds on the Power Virtual Server with VPC landing zone to provide a fully automated process to build an SAP HANA workspace in Power Virtual Server. There are two variations available for Power Virtual Server for SAP HANA:

1. Variation 'SAP ready Power Virtual Server' creates an SAP-tuned HANA and NetWeaver configuration to IBM Power Virtual Server hosts.
2. Variation 'SAP S/4HANA or BW/4HANA' providing SAP S/4HANA or SAP BW/4HANA installation configuration to IBM Power Virtual Server hosts.

Figure 4-1 shows the different variations and the actions that each variation completes

Variation	Available on IBM Catalog	Requires Schematics Workspace ID	Creates PowerVS with VPC landing zone	Creates PowerVS HANA Instance	Creates PowerVS NW Instances	Performs PowerVS OS Config	Performs PowerVS SAP Tuning	Install SAP software
IBM catalog PowerVS SAP Ready	✓	✓	N/A	1	0 to N	✓	✓	N/A
IBM catalog SAP S/4HANA or BW/4HANA variation	✓	✓	N/A	1	1	✓	✓	✓

Figure 4-1 Variations of Power Virtual Server for SAP HANA

Figure 4-2 shows the IBM catalog entries for the two variants.

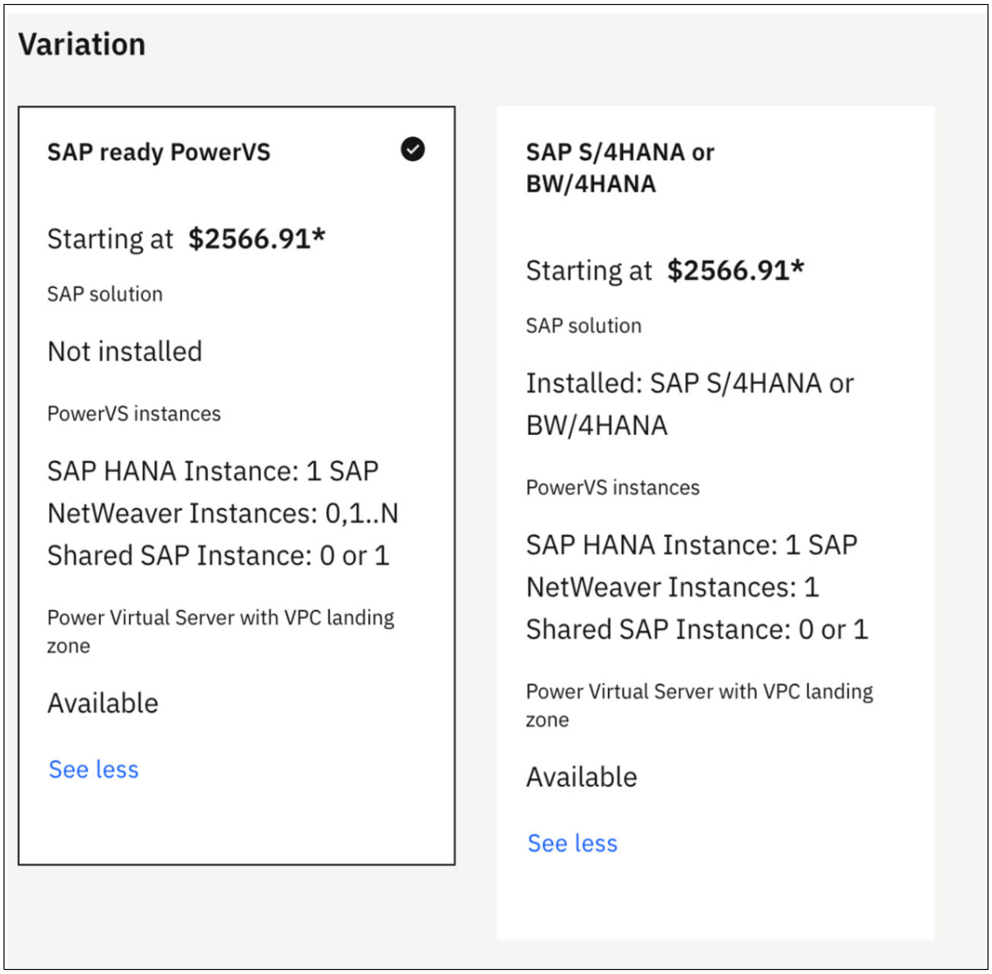


Figure 4-2 Two available variations for Power Virtual Server for SAP HANA

4.2.1 Variant SAP ready Power Virtual Server

This Power Virtual Server variation streamlines the creation of a basic, yet expandable, SAP system landscape. Building upon the foundation of the Power Virtual Server with VPC landing zone, it deploys and preconfigures Power Virtual Server instances for SAP HANA, SAP NetWeaver, and optional shared SAP files.

In addition, the deployable architecture creates the network connections required for connectivity and other networking components such as proxy servers and shared NFS services.

Important: This variant focuses on infrastructure provisioning and does not include SAP software installation. Clients are responsible for deploying SAP applications on the prepared infrastructure.

By using this deployable architecture, organizations can accelerate their SAP deployments on IBM Power Virtual Server and benefit from a robust, scalable, and secure SAP landscape. Figure 4-3 on page 50 shows the shows the components created by this version of the deployable architecture.

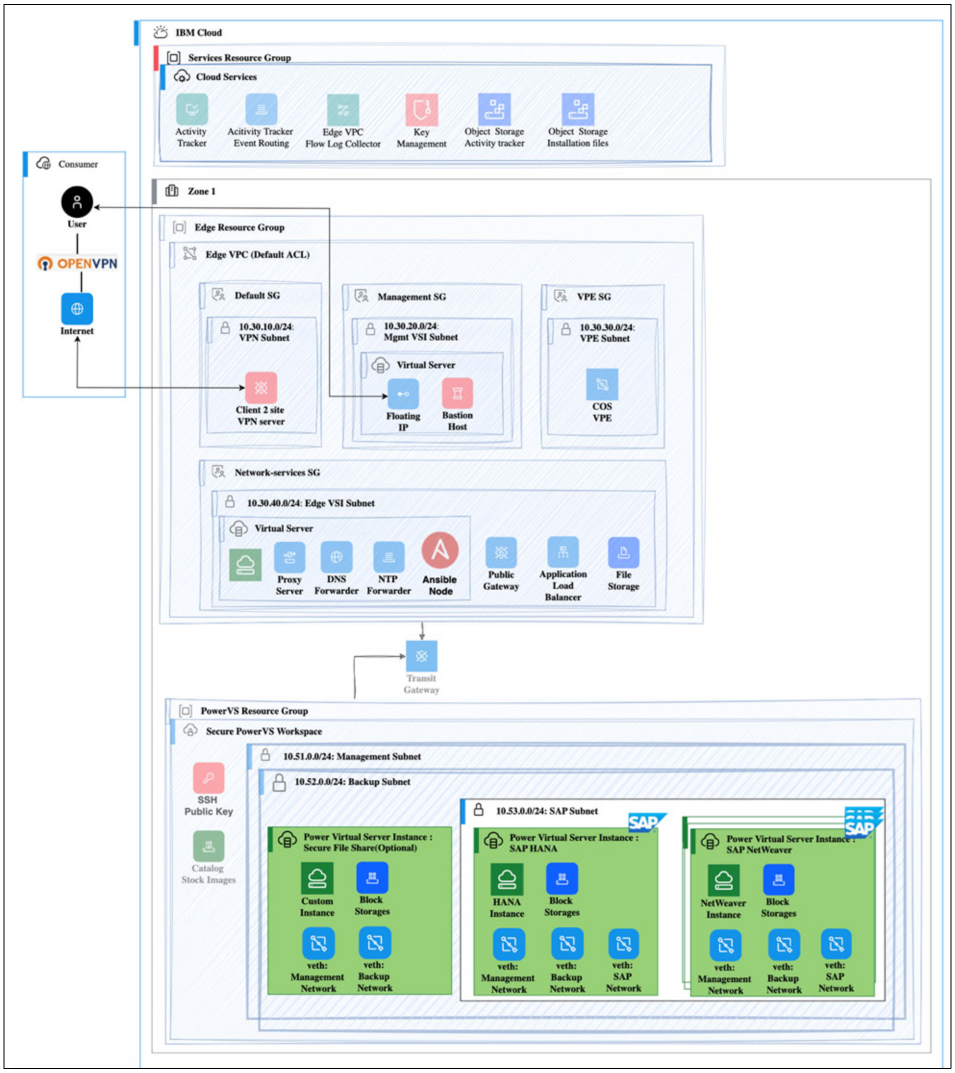


Figure 4-3 Power Virtual Server instances prepared to run SAP

Overview of tasks executed by this deployable architecture:

- Creates a new private subnet for SAP communication for the entire landscape.
- Creates and configures one Power Virtual Server instance for SAP HANA based on best practices.
- Creates and configures multiple Power Virtual Server instances for SAP NetWeaver based on best practices.
- Creates and configures one optional Power Virtual Server instance that can be used for sharing SAP files between other system instances.
- Connects all created Power Virtual Server instances to a proxy server specified by IP address or hostname.
- Optionally connects all created Power Virtual Server instances to an NTP server and DNS forwarder specified by IP address or hostname.
- Optionally configures a shared NFS directory on all created Power Virtual Server instances.
- For post-instance provisioning the *power_linux_sap* role from Ansible Galaxy is executed.

- Tested with RHEL8.4/8.6/8.8/9.2, SLES15-SP3/SP5 images.

> **Important:** As a prerequisite the Sta**ndard Variation** of the *Power Virtual Server with VPC landing zone* deployable architecture needs to be created first.

Notes

Take note of the following important considerations when implementing this deployable architecture:

- No SAP software or solutions are installed.
- The HANA data and HANA log filesystems sizes are calculated automatically based on the memory size.
- A custom storage configuration by providing custom volume size, iops (tier0, tier1, tier3, tier5k), counts and mount points is supported.
- All filesystems provisioned for sharefs instance will be NFS exported and mounted on all NetWeaver Instances, if sharefs instance is enabled.
- Do not specify a filesystem /sapmnt explicitly for the NetWeaver instance as it is created internally when sharefs instance is not enabled.

The reference architecture for Power Virtual Server for SAP HANA variation SAP ready Power Virtual Server is designed to provide Power Virtual Server Linux instances prepared and configured for SAP HANA and SAP NetWeaver workloads according to the best practices and requirements using IBM Cloud deployable architectures framework.

4.2.2 Variation - SAP S/4HANA or BW/4HANA

This variation of the Power Virtual Server for SAP HANA creates a basic and expandable SAP system landscape. The variation builds on the foundation of Power Virtual Server with VPC landing zone deployable architecture. Power Virtual Server instances for SAP HANA, SAP NetWeaver and optionally for shared SAP files are deployed and preconfigured for SAP installation.

Figure 4-4 shows the components created by this version of the deployable architecture.

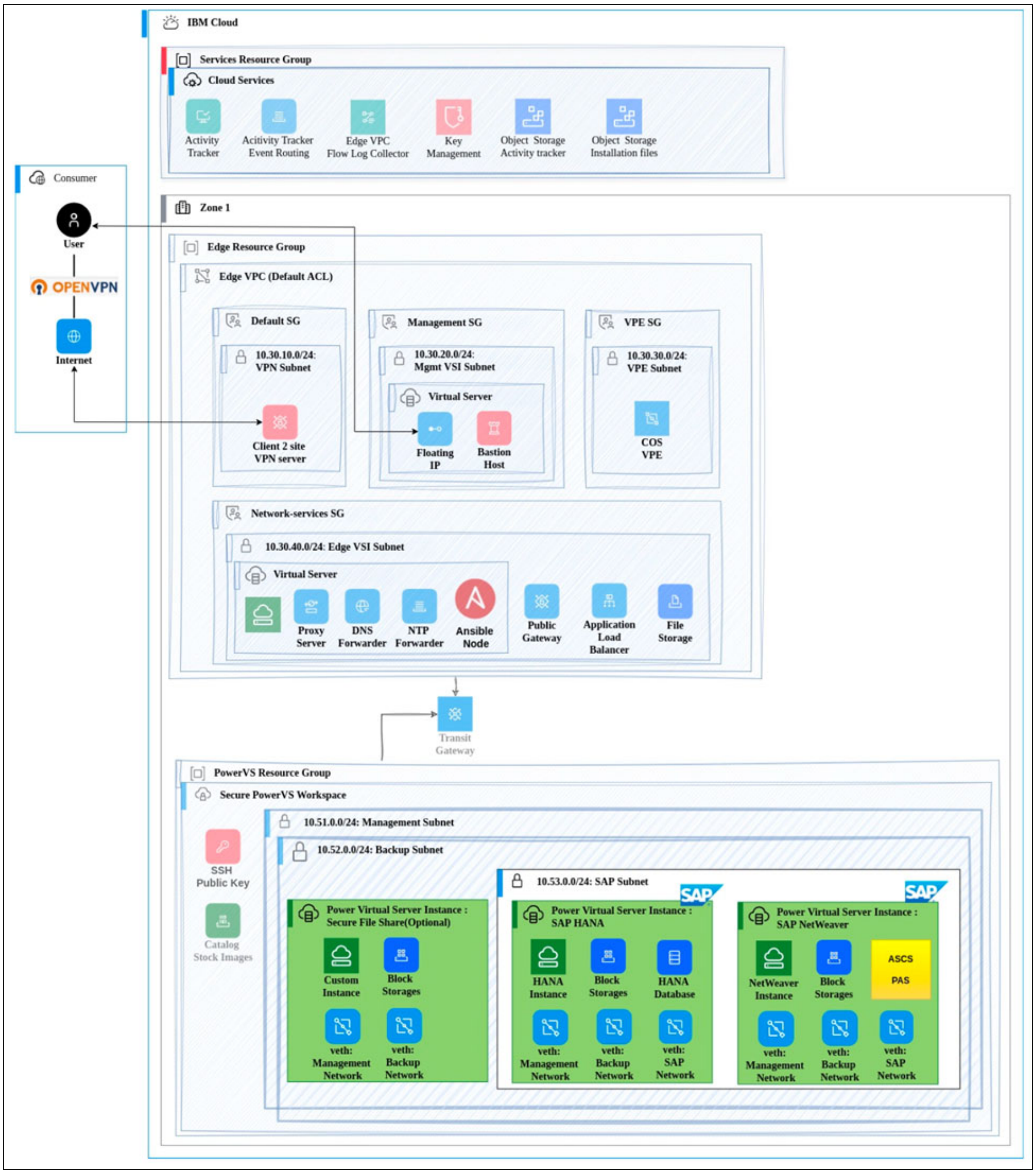


Figure 4-4 Power Virtual Server instances prepared to run SAP

As a result of running this deployable architecture variation, the S/4HANA or BW/4HANA solution is installed based on the selected version. Services such as DNS, NTP and NFS running in the IBM Cloud VPC and provided by Power Virtual Server with VPC landing zone are leveraged. The transit gateway provides the network bridge between the IBM Power infrastructure and the IBM VPC and public internet.

The resulting SAP landscape leverages the services such as Activity Tracker, Cloud Object Storage, Key Management from the VPC landing zone and the network connectivity configuration provided by Power Virtual Server with VPC landing zone.

Overview of tasks executed by this deployable architecture

- Creates a new private subnet for SAP communication for the entire landscape.
- Creates and configures one Power Virtual Server instance for SAP HANA based on best practices for HANA database.
- Creates and configures one Power Virtual Server instance for SAP NetWeaver based on best practices, hosting the PAS and ASCS instances.
- Creates and configures one optional Power Virtual Server instance for sharing SAP files between other system instances.
- Connects all created Power Virtual Server instances to a proxy server specified by IP address or hostname.
- Optionally connects all created Power Virtual Server instances to an NTP server and DNS forwarder specified by IP address or hostname.
- Optionally configures a shared NFS directory on all created Power Virtual Server instances.
- Supports installation of S/4HANA2023, S/4HANA2022, S/4HANA2021, S/4HANA2020, BW/4HANA2021.

Take note of the following important considerations when implementing this variation of the deployable architecture:

- The HANA data and HANA log filesystems sizes are calculated automatically based on the memory size.
- A custom storage configuration by providing custom volume size, iops (tier0, tier1, tier3, tier5k), counts and mount points is supported.
- All filesystems provisioned for sharefs instance will be NFS exported and mounted on all NetWeaver Instances, if sharefs instance is enabled.
- Do not specify a filesystem /sapmnt explicitly for the NetWeaver instance as it is created internally when sharefs instance is not enabled.

Important: This deployable architecture has as a prerequisite the Power Virtual Server with VPC landing zone. For this variation, you also need to have created a Cloud Object Storage Bucket containing the SAP Software installation media files.

4.3 Considerations and planning

The following sections discuss specific prerequisite steps that need to be taken prior to running this deployable architecture

4.3.1 Power Virtual Server for SAP HANA SAP Ready

The outcome of this deployable architecture is an SAP-tuned HANA and NetWeaver configuration on IBM Power Virtual Server hosts.

The Power Virtual Server for SAP HANA SAP Ready deployable architecture completes the following tasks:

- Creates a new private subnet for SAP communication for the entire landscape.
- Creates and configures one Power Virtual Server instance for SAP HANA based on best practices.

- Creates and configures multiple Power Virtual Server instances for SAP NetWeaver based on best practices.
- Creates and configures one optional Power Virtual Server instance that can be used for sharing SAP files between other system instances.
- Connects all created Power Virtual Server instances to a proxy server specified by IP address or hostname.
- Optionally connects all created Power Virtual Server instances to an NTP server and DNS forwarder specified by IP address or hostname.
- Optionally configures a shared NFS directory on all created Power Virtual Server instances.
- During post-instance provisioning the Ansible Galaxy collection role power_linux_sap from IBM is executed.

This deployable architecture has been tested with RHEL8.4, RHEL8.6, RHEL8.8, and RHEL9.2 images as well as SLES15 SP3 and SP5 images.

Important: Does not install any SAP software or solutions.

Note that:

- Filesystem sizes for HANA data and HANA log are calculated automatically based on the memory size.
- Custom storage configuration by providing custom volume size, iops (tier0, tier1, tier3, tier5k), counts, and mount points is supported.
- If sharefs instance is enabled, then all filesystems provisioned for sharefs instance will be NFS exported and mounted on all NetWeaver Instances.
- Do not specify a filesystem */sapmnt* explicitly for NetWeaver instance as it is created internally when sharefs instance is not enabled.

Important: This solution requires a *schematics workspace ID* as input before you begin. This is created using the Power Virtual Server with VPC landing zone deployable architecture prior to deployment

4.3.2 Power Virtual Server for SAP HANA SAP S/4HANA or SAP BW/4HANA'

The outcome of this deployable architecture is an SAP S/4HANA or SAP BW/4HANA installation configuration on IBM Power Virtual Server hosts.

The Power Virtual Server for SAP HANA - SAP S/4HANA or SAP BW/4HANA deployable architecture completes the following tasks:

- Creates a new private subnet for SAP communication for the entire landscape.
- Creates and configures one Power Virtual Server instance for SAP HANA based on best practices for HANA database.
- Creates and configures one Power Virtual Server instance for SAP NetWeaver based on best practices, hosting the PAS and ASCS instances.
- Creates and configures one optional Power Virtual Server instance for sharing SAP files between other system instances.
- Connects all created Power Virtual Server instances to a proxy server specified by IP address or hostname.

- Optionally connects all created Power Virtual Server instances to an NTP server and DNS forwarder specified by IP address or hostname.
- Optionally configures a shared NFS directory on all created Power Virtual Server instances.

Supports installation of S/4HANA2023, S/4HANA2022, S/4HANA2021, S/4HANA2020, and BW/4HANA2021. Also Supports installation using *Maintenance Planner.*

Note that:

- Filesystem sizes for HANA data and HANA log are calculated automatically based on the memory size.
- Custom storage configuration by providing custom volume size, iops (tier0, tier1, tier3, tier5k), counts, and mount points is supported.
- If sharefs instance is enabled, then all filesystems provisioned for sharefs instance will be NFS exported and mounted on all NetWeaver Instances.
- Do not specify a filesystem */sapmnt* explicitly for NetWeaver instance as it is created internally when sharefs instance is not enabled.

Important: This solution requires a schematics workspace ID as input before you begin. This is created using the Power Virtual Server with VPC landing zone deployable architecture prior to deployment.

Prerequisites

1. IBM Cloud Object Storage service credentials are required.

   Recommended to have a COS instance in the same region where the S/4HANA or BW/4HANA deployment is planned, as copying the files onto the LPAR will be faster.

   The '*ibmcloud_cos_service_credentials*' variable requires a value in JSON format. This can be obtained using the instructions here.
2. The SAP binaries required for installation need to be uploaded to an IBM Cloud Object Storage bucketin the correct folder structure.

   All binaries for HANA database and SAP solution (S/4HANA or BW/4HANA) must be uploaded to the IBM Cloud Object Storage Instance bucket in IBM Cloud before starting this deployment.

   For example the binaries required for S/4HANA 2023 and BW/4HANA 2021 are listed here.

   Example 4-1 provide an example of the folder structure.

Example 4-1 Folder structure

```
S4HANA_2023
|
|_HANA_DB
| |_all IMDB* Files and SAPCAR files
|
|_S4HANA_2023
  |
```

Important: Do not mix the HANA DB binaries with the S/4HANA or BW/4HANA solution binaries otherwise the Ansible playbook execution will fail.

- If you have a maintenance planner stack XML file, place it under the same folder as S4HANA_2023 and not under the HANA DB directory. Applies to all other versions as well. Place the name of this file in **'cos_swpm_mp_stack_file_name'**. Leave it empty if you do not have this stack XML file.
- The '*ibmcloud_cos_configuration*' variable must be set correctly based on the folder structure created.
  - `cos_region`: region of IBM Cloud Object Storage instance bucket. Example: **eu-gb**
  - `cos_bucket_name`: cos bucket name
  - `cos_hana_software_path`: folder path to HANA db binaries from the root of the bucket.
  - `cos_solution_software_path`: folder path to S/4HANA binaries from the root of the bucket. Example from point 3, the value would be: **"s4hana2023/S4HANA_2023"**
  - `cos_swpm_mp_stack_file_name`: Stack XML file name. Value must be set to empty (`' '`) if not available. If value is provided, then this file **must be present** in the same path as `'cos_solution_software_path'`.

Post Deployment

- All the installation logs and Ansible playbook files will be under the directory: `/root/terraform_files/`.
- The *ansible vault password* will be used to encrypt the Ansible playbook file created during deployment. This playbook file will be placed under:
  - `/root/terraform_files/sap-hana-install.yml` on the HANA instance
  - `/root/terraform_files/sap-swpm-install-vars.yml` on the NetWeaver Instance.
- This file can be decrypted using the same value passed to variable `'ansible_vault_password'` during deployment. Use the command:

  ```
  ansible-vault decrypt /root/terraform_files/sap-swpm-install-vars.yml
  ```

  and enter the password when prompted.

Storage setup

The following options are used to configure the storage for the Power Virtual Server instances.

1. HANA Instance

   The storage for the HANA instance is created using either the default values or by using the optional variables for customized sizes. The default values are:

   - `/hana/shared (size auto calculated based on memory)`
   - `/hana/data   (size auto calculated based on memory)`
   - `/hana/log    (size auto calculated based on memory)`
   - `/usr/sap     50GB`

2. NetWeaver Instance

   Like the HANA instance , the NetWeaver instance storage is created using either default values or custom values defined by the provided optional variables. The default values

   - `/usr/sap 50GB`
   - `/sapmnt  300GB (only if sharefs instance is not provisioned)`

3. Sharefs Instance

   A shared filesystem is defined using either default values or values specified in the provided optional variables. The default values are:

   - `/sapmnt    300GB`
   - `/usr/trans 50GB`

Ansible roles used

The following Ansible roles for SAP are used,

RHEL System Roles:

- `sap_hana_install`
- `sap_swpm`
- `sap_general_preconfigure`
- `sap_hana_preconfigure`
- `sap_netweaver_preconfigure`

IBM Role:

- `power_linux_sap`

4.3.3 Deployment dependencies

The SAP solution for end-to-end deployment by using deployable architectures is logically separated into the following steps:

As a first step it is mandatory to have an existing deployment of IBM Power Virtual Server with VPC landing zone deployable architecture. This can be deployed from the IBM catalog.

The three supported variations of IBM Power Virtual Server with VPC landing zone are:

- Create a new architecture: Standard Variation
- Extend IBM Power Virtual Server with VPC landing zone: Standard Variation
- Create a new architecture: Import Variation

If you already have a Power Virtual Server workspace, then you can use the Import Power Virtual Server workspace variation of the IBM Power Virtual Server with VPC landing zone deployable architecture instead of deploying an entirely new one. The import variation creates a schematics workspace by taking pre-existing VPC and Power Virtual Server infrastructure resource details as inputs. The ID of this schematics workspace will be the pre-requisite workspace id required by the IBM Power Virtual Server for SAP HANA to create and configure the Power Virtual Server instances for SAP on top of the existing infrastructure.

The second step is deploying IBM Power Virtual Server for SAP HANA on top of IBM Power Virtual Server with VPC landing zone.

Note: Multiple SAP solutions (systems) can be deployed in one IBM Power Virtual Server workspace that uses this deployable architecture if the private CIDR network range of each SAP solution does not overlap.

Note: The second step relies on the previous one. The prerequisite dependency is constrained by type. This dependency means that not every Power Virtual Server workspace is accepted for the IBM Power Virtual Server for SAP HANA deployable architecture, but only the one created by the IBM Power Virtual Server with VPC landing zone since its Schematics workspace ID is a pre-requisite for the IBM Power Virtual Server for SAP HANA to create and configure the Power Virtual Server instances for SAP on top of the existing infrastructure.

4.3.4 Before you begin deploying

There are six necessary steps required to enable a user to successfully deploy the architecture:

Step 1: Setting the IAM Permissions

Set up account access (Cloud Identity and Access Management (IAM)):

– Create an IBM Cloud API key

. The user who owns this key must have the *Administrator* role.

– For compliance with IBM Cloud Framework for Financial Services you should require users in your account to use multifactor authentication (MFA).

– Set up access groups.

– User access to IBM Cloud resources is controlled by using the access policies that are assigned to access groups. For IBM Cloud Financial Services validation, do not assign direct IAM access to any IBM Cloud resources.

– Select *All Identity and Access enabled services* when you assign access to the group.

Verify the access roles

IAM access roles are required to install this deployable architecture and create all the required elements.

The following permissions are needed for this deployable architecture:

– Create services from IBM Cloud catalog.
– Create and modify IBM Cloud VPC services, virtual server instances, networks, network prefixes, storage volumes, SSH keys, and security groups of this VPC.
– Create and modify Power Virtual Server services, virtual server instances, networks, storage volumes, ssh keys of this Power Virtual Server.
– Create and modify IBM Cloud direct links and IBM Cloud Transit Gateway.
– Access existing Object Storage services.

Access for IBM Cloud projects

You can use IBM Cloud projects as a deployment option. Projects are designed with infrastructure as code and compliance in mind to help ensure that your projects are managed, secure, and always compliant.

The following access is needed to create a project and create project tooling resources within the account:

– The Editor role on the Projects service.
– The Editor and Manager role on the Schematics service.
– The Viewer role on the resource group for the project.

Step 2: IBM Power Virtual Server with VPC landing zone deployable architecture

An existing deployment of the IBM Power Virtual Server with VPC landing zone deployable architecture must be available.

Step 3: Define hostnames

Define hostnames for all the SAP services that will be deployed in the landscape. The hostnames will be mapped later to the IP addresses after the resources are provisioned. The domain name must be the same for all hostnames that belong to one SAP system.

Step 4: Determine size parameters

The size parameters need to be determined for each SAP workload that will be deployed. The most important parameters are memory size (especially for SAP HANA) and the number of SAPs. Most of the other configurations can be derived from these two key metrics.

More information about the SAP sizing process can be seen at Sizing process for SAP Systems . Also see SAP Sizing.

For deployment of each SAP system, you must be aware about how many SAP NetWeaver instances you plan to deploy. Decide whether you plan to use a separate Power Virtual Server instance for hosting SAP shared files.

For deployment of each SAP NetWeaver Power Virtual Server instance, you must be aware of the following sizing parameters:

- Memory size
- Number of CPUs

For deployment of SAP HANA Power Virtual Server instance, you must be aware of the following sizing parameters:

- SAP t-shirt size as combination of memory and number of CPUs. Certified profiles can be checked here.

Step 5: SAP software installation packages for the SAP S/4HANA or BW/4HANA variation only

> **Important:** The following steps only apply to the SAP S/4HANA or BW/4HANA variation. You must have an existing IBM Cloud Object Storage instance.

- Within the instance, an Object Storage bucket that contains the SAP Software installation media files is required.
- To configure access to these software packages from a running virtual server instance, you need the following information:
  - Object Storage endpoint
  - Object Storage bucket name
  - Object Storage HMAC access key
  - Object Storage HMAC secret access key

Section 4.3.5, “How to create a COS instance for SAP installation file” on page 59 details the steps to create a Cloud Object Storage (COS) instance and how to organize the SAP binaries in the correct structure.

Step 6: Learn about deployment input parameters

Ensure that you are familiar with the required input for the deployment execution. See section 4.4, “Inputs” on page 61 for input parameters.

4.3.5 How to create a COS instance for SAP installation file

This deployable architecture installs the SAP binaries needed to run your SAP instance. The files used for installation need to be acquired from SAP and are stored in a Cloud Object Store bucket to be accessed when the deployable architecture is run.

The IBM Power Virtual Server with VPC landing zone comes with a storage bucket that is meant to be used for the installation files. If you have deployed an instance without creating the COS instance, these instructions will assist you in creating the COS instance before running the deployable architecture.

Getting Stared

The following steps will assist you in creating a COS instance and placing the SAP installation binaries in the bucket in the proper format for installing SAP.

SAP Installation Binaries

HANA and NetWeaver installation binaries need to be obtained from SAP. The required installation files depend on the version you are trying to install. Check out the Power Virtual Server SAP repo and the sap_hana_install community role to identify the needed files.

Step 1: Create a COS Service Instance with a Storage Bucket

Before you can create a storage bucket to store the SAP installation files, you need to create a COS service instance. The service instance is where your storage buckets reside.

1. Create a Cloud Object Storage service instance

   Navigate to **Cloud Object Storage**:

   - Click the **Navigation menu icon** → **Classic Infrastructure** → **Object Storage**
   - Click **Create an instance**

   Configure a Cloud Object Storage service instance

   - Under **Choose an Infrastructure**, select **IBM Cloud**.
   - Select a pricing plan.
   - Configure service name, resource group, and tags.
   - Click **Create**. It will create a service instance and take you to it.

2. Configure Service Credentials

   You need to create service credentials to enable the deployable architecture to log into the COS instance. You also need them if you want to setup your bucket via the CLI. The deployable architecture requires HMAC credentials, you can find more information here.

   – Navigate to your service instance. You can find it by clicking on the **Navigation menu icon** → **Resource List** → **Storage** → **your COS service instance Name**

   – In the service instance, select the **Service credentials** tab on the top.

   – Click **New Credential**.

   - Give it a name
   - For role, select **Manager**
   - For Service ID, select **Create New Service id** and remember what you name it
   - Make sure you enable **Include HMAC Credential**

3. Create a Storage Bucket:

   Navigate to your service instance. The previous step should have taken you directly to your service instance. If it didn't, you can find it by clicking on the **Navigation menu icon** → **Resource List** → **Storage** → **your COS service instance Name**

   – Select **Create a Custom Bucket.**
   – Select a **Name** and **Location** close to your deployment.
   – For the other settings, you can leave the default settings or refer to the Object Storage Documentation for more information about the configuration options and encryption.
   – Once done, click **Create Bucket**

4. Configure Bucket Access Policies

   Navigate to your bucket. You can find it in the **Buckets** tab in your service instance.

   Inside your bucket, go to the Permissions tab and open Access policies.

   – For **Policy** type, select **Service ID**
   – For **Select a service ID**, select the **service id generated in the previous step**
   – For **Role for this bucket**, select **Manager**
   – Click **Create access policy**

Step 2: Upload SAP installation binaries

1. Place the SAP installation binaries in the same folder structure you'd like in your bucket.
2. Place all your HANA DB files in one directory and all your S/4HANA or BW/4HANA in another one. Do not mix the HANA DB binaries with the S/4HANA or BW/4HANA solution binaries in the same folder, otherwise the ansible playbook execution will fail.
3. Upload the files

   Navigate to your bucket like described in the previous steps.

   – Click **Upload**, a new menu should pop up.
   – Select your folder and hit **Upload**.

4.4 Inputs

The input variables differ between the different variations as the objects created are different in each variant.

Table 4-1 shows the input variables for the Power Virtual Server for SAP HANA - SAP Ready Power Virtual Server variation.

Table 4-1 Input variables for Power Virtual Server for SAP HANA - SAP Ready

Name	Description	Type	Default	Req
ibmcloud_api_key	IBM Cloud platform API key needed to deploy IAM enabled resources.	string	n/a	yes
nfs_directory	Target directory on which the file storage share from VPC will be mounted.	string	"/nfs"	no
os_image_distro	Image distribution to use for all instances(Shared, HANA, NetWeaver). OS release versions may be specified in 'powervs_sap_default_images' optional parameters below.	string	n/a	yes
powervs_create_separate_sharefs_instance	Deploy separate IBM Power Virtual Server instance as central file system share. µŸ All filesystems defined in 'powervs_sharefs_instance_storage_config' variable will be NFS exported and mounted on SAP NetWeaver Power Virtual Server instances if enabled. µŸ Optional parameter 'powervs_share_fs_instance' can be configured if enabled.	bool	n/a	yes

Name	Description	Type	Default	Req

powervs_default_sap_images	Default SUSE and Red Hat Linux images to use for Power Virtual Server SAP HANA and SAP NetWeaver instances.	object({
		sles_hana_image = string
		sles_nw_image = string
		rhel_hana_image = string
		rhel_nw_image = string
		})
			{
			"rhel_hana_image": "RHEL9-SP2-SAP",
			"rhel_nw_image": "RHEL9-SP2-SAP-NETWEAVER",
			"sles_hana_image": "SLES15-SP5-SAP",
			"sles_nw_image": "SLES15-SP5-SAP-NETWEAVER"
			}
				no

powervs_hana_instance_additional_storage_config	Additional File systems to be created and attached to Power Virtual Server SAP HANA instance.
	µŸ 'size' is in GB.
	µŸ 'count' specify over how many storage volumes the file system will be striped.
	µŸ 'tier' specifies the storage tier in Power Virtual Server workspace.
	µŸ 'mount' specifies the target mount point on OS.
		list(object({
		name = string
		size = string
		count = string
		tier = string
		mount = string
		}))
			[
			{
			"count": "1",
			"mount": "/usr/sap",
			"name": "usrsap",
			"size": "50",
			"tier": "tier3"
			}
			]
				no

powervs_hana_instance_custom_storage_config	Custom file systems to be created and attached to Power Virtual Server SAP HANA instance.
	µŸ 'size' is in GB.
	µŸ 'count' specify over how many storage volumes the file system will be striped.
	µŸ 'tier' specifies the storage tier in Power Virtual Server workspace.
	µŸ 'mount' specifies the target mount point on OS.
		list(object({
		name = string
		size = string
		count = string
		tier = string
		mount = string
		pool = optional(string)
		}))
			[
			{
			"count": "",
			"mount": "",
			"name": "",
			"size": "",
			"tier": ""
			}
			]
				no

powervs_hana_instance_name	Power Virtual Server SAP HANA instance hostname (non FQDN). µŸ Will get the form of -. µŸ Max length of final hostname must be <= 13 characters.	string	"hana"	no
powervs_hana_instance_sap_profile_id	Power Virtual Server SAP HANA instance profile to use. Must be one of the supported profiles. µŸ File system sizes are automatically calculated. µŸ Override automatic calculation by setting values in optional parameter 'powervs_hana_instance_custom_storage_config'.	string	"ush1-4x256"	no
powervs_netweaver_cpu_number	Number of CPUs for each Power Virtual Server SAP NetWeaver instance.	string	"3"	no
powervs_netweaver_instance_count	Number of Power Virtual Server SAP NetWeaver instances that should be created.	number		no
powervs_netweaver_instance_name	Power Virtual Server SAP NetWeaver instance hostname (non FQDN). µŸ Will get the form of --. µŸ Max length of final hostname must be <= 13 characters.	string	"nw"	no

Name	Description	Type	Default	Req

powervs_netweaver_instance_storage_config	File systems to be created and attached to Power Virtual Server SAP NetWeaver instance.
	µŸ 'size' is in GB.
	µŸ 'count' specify over how many storage volumes the file system will be striped.
	µŸ 'tier' specifies the storage tier in Power Virtual Server workspace.
	µŸ 'mount' specifies the target mount point on OS.
	µŸ Do not specify volume for 'sapmnt' as this will be created internally if 'powervs_create_separate_sharefs_instance' is false, else 'sapmnt' be will mounted from sharefs instance.
		list(object({
		name = string
		size = string
		count = string
		tier = string
		mount = string
		pool = optional(string)
		}))
			[
			{
			"count": "1",
			"mount": "/usr/sap",
			"name": "usrsap",
			"size": "50",
			"tier": "tier3"
			}
			]
				no

powervs_netweaver_memory_size	Memory size for each Power Virtual Server SAP NetWeaver instance.	string	"32"	no
powervs_sap_network_cidr	Network range for dedicated SAP network. Used for communication between SAP Application servers with SAP HANA Database. E.g., '10.53.0.0/24'	string	"10.53.0.0/24"	no

powervs_sharefs_instance	Share fs instance.
	µŸ This parameter is effective if 'powervs_create_separate_sharefs_instance' is set to true.
	µŸ size' is in GB.
	µŸ 'count' specify over how many storage volumes the file system will be striped.
	µŸ 'tier' specifies the storage tier in Power Virtual Server workspace.
	µŸ 'mount' specifies the target mount point on OS.
		object({
		name = string
		processors = string
		memory = string
		proc_type = string
		storage_config = list(object({
		name = string
		size = string
		count = string
		tier = string
		mount = string
		pool = optional(string)
		}))
		})
			{
			"memory": "2",
			"name": "share",
			"proc_type": "shared",
			"processors": "0.5",
			"storage_config": [
			{
			"count": "1",
			"mount": "/sapmnt",
			"name": "sapmnt",
			"size": "300",
			"tier": "tier3"
			},
			{
			"count": "1",
			"mount": "/usr/trans",
			"name": "trans",
			"size": "50",
			"tier": "tier3"
			}
			]
			}
				no

powervs_zone	IBM Cloud data center location corresponding to the location used in 'Power Virtual Server with VPC landing zone' pre-requisite deployment.	string	n/a	yes
prefix	Unique prefix for resources to be created (e.g., SAP system name). µŸ Max length must be less than or equal to 6.	string	n/a	yes
prerequisite_workspace_id	IBM Cloud Schematics workspace ID of an existing 'Power Virtual Server with VPC landing zone' catalog solution.	string	n/a	yes
sap_domain	SAP network domain name.	string	"sap.com"	no

Name	Description	Type	Default	Req
ssh_private_key	Private SSH key (RSA format) used to login to IBM Power Virtual Server instances. µŸ Should match to uploaded public SSH key referenced by 'ssh_public_key' which was created previously. ► The key is temporarily stored and deleted.	string	n/a	yes

Table 4-2 shows the input variables for the Power Virtual Server for SAP HANA - SAP S/4HANA or SAP BW/4HANA,

Table 4-2 Input variables for Power Virtual Server for SAP HANA - SAP S/4HANA or BW/4HANA

Name	Description	Type	Default	Req
ansible_vault_password	Vault password to encrypt SAP installation parameters in the OS.	string	n/a	yes
ibmcloud_api_key	IBM Cloud platform API key needed to deploy IAM enabled resources.	string	n/a	yes

ibmcloud_cos_configuration	Cloud Object Storage instance containing SAP installation files that will be downloaded to NFS share.
	µŸ 'cos_hana_software_path' must contain only binaries required for HANA DB installation.
	µŸ 'cos_solution_software_path' must contain only binaries required for S/4HANA or BW/4HANA installation and must not contain any IMDB files.
	µŸ If you have an optional stack xml file (maintenance planner), place it under the 'cos_solution_software_path' directory.
	µŸ Avoid inserting '/' at the beginning for 'cos_hana_software_path' and 'cos_solution_software_path'.
		object({
		cos_region = string
		cos_bucket_name = string
		cos_hana_software_path = string
		cos_solution_software_path = string
		cos_swpm_mp_stack_file_name = string
		})
			{
			"cos_bucket_name": "powervs-automation",
			"cos_hana_software_path": "HANA_DB/rev66",
			"cos_region": "eu-geo",
			"cos_solution_software_path": "S4HANA_2022",
			"cos_swpm_mp_stack_file_name": ""
			}
				no

ibmcloud_cos_service_credentials	IBM Cloud Object Storage instance service credentials to access the bucket in the instance.	string	n/a	yes
powervs_create_separate_sharefs_instance	Deploy separate IBM Power Virtual Server instance as central file system share. µŸ All filesystems defined in 'powervs_sharefs_instance_storage_config' variable will be NFS exported and mounted on SAP NetWeaver Power Virtual Server instances if enabled. µŸ Optional parameter 'powervs_share_fs_instance' can be configured if enabled.	bool	n/a	yes

powervs_default_sap_images	Default Red Hat Linux images to use for Power Virtual Server SAP HANA and SAP NetWeaver instances.	object({
		rhel_hana_image = string
		rhel_nw_image = string
		})
			{
			"rhel_hana_image": "RHEL9-SP2-SAP",
			"rhel_nw_image": "RHEL9-SP2-SAP-NETWEAVER"
			}
				no

Name	Description	Type	Default	Req

powervs_hana_instance_additional_storage_config	Additional File systems to be created and attached to Power Virtual Server SAP HANA instance.
	µŸ 'size' is in GB.
	µŸ 'count' specify over how many storage volumes the file system will be striped.
	µŸ 'tier' specifies the storage tier in Power Virtual Server workspace.
	µŸ 'mount' specifies the target mount point on OS.
		list(object({
		name = string
		size = string
		count = string
		tier = string
		mount = string
		}))
			[
			{
			"count": "1",
			"mount": "/usr/sap",
			"name": "usrsap",
			"size": "50",
			"tier": "tier3"
			}
			]
				no

powervs_hana_instance_custom_storage_config	Custom file systems to be created and attached to Power Virtual Server SAP HANA instance.
	µŸ 'size' is in GB.
	µŸ 'count' specify over how many storage volumes the file system will be striped.
	µŸ 'tier' specifies the storage tier in Power Virtual Server workspace.
	µŸ 'mount' specifies the target mount point on OS.
		list(object({
		name = string
		size = string
		count = string
		tier = string
		mount = string
		}))
			[
			{
			"count": "",
			"mount": "",
			"name": "",
			"size": "",
			"tier": ""
			}
			]
				no

powervs_hana_instance_name	Power Virtual Server SAP HANA instance hostname (non FQDN). µŸ Max length of final hostname must be <= 13 characters.	string	"hana"	no
powervs_hana_instance_sap_profile_id	Power Virtual Server SAP HANA instance profile to use. Must be one of the supported profiles. µŸ File system sizes are automatically calculated. µŸ Override automatic calculation by setting values in optional parameter 'powervs_hana_instance_custom_storage_config'.	string	"ush1-4x256"	no
powervs_netweaver_cpu_number	Number of CPUs for Power Virtual Server SAP NetWeaver instance.	string	"3"	no
powervs_netweaver_instance_name	Power Virtual Server SAP NetWeaver instance hostname (non FQDN). µŸ Max length of final hostname must be <= 13 characters.	string	"nw"	no

powervs_netweaver_instance_storage_config	File systems to be created and attached to Power Virtual Server SAP NetWeaver instance.
	µŸ 'size' is in GB.
	µŸ 'count' specify over how many storage volumes the file system will be striped.
	µŸ 'tier' specifies the storage tier in Power Virtual Server workspace.
	µŸ 'mount' specifies the target mount point on OS.
	µŸ Do not specify volume for 'sapmnt' as this will be created internally if 'powervs_create_separate_sharefs_instance' is false, else 'sapmnt' will mounted from sharefs instance.
		list(object({
		name = string
		size = string
		count = string
		tier = string
		mount = string
		}))
			[
			{
			"count": "1",
			"mount": "/usr/sap",
			"name": "usrsap",
			"size": "50",
			"tier": "tier3"
			}
			]
				no

powervs_netweaver_memory_size	Memory size for Power Virtual Server SAP NetWeaver instance.	string	"32"	no

Name	Description	Type	Default	Req
powervs_sap_network_cidr	Network range for dedicated SAP network. µŸ Used for communication between SAP Application servers with SAP HANA Database. E.g., '10.53.0.0/24'	string	"10.53.0.0/24"	no

powervs_sharefs_instance	Share fs instance.
	µŸ This parameter is effective if 'powervs_create_separate_sharefs_instance' is set to true.
	µŸ 'size' is in GB.
	µŸ 'count' specify over how many storage volumes the file system will be striped.
	µŸ 'tier' specifies the storage tier in Power Virtual Server workspace.
	µŸ 'mount' specifies the target mount point on OS.
		object({
		name = string
		processors = string
		memory = string
		proc_type = string
		storage_config = list(object({
		name = string
		size = string
		count = string
		tier = string
		mount = string
		}))
		})
			{
			"memory": "2",
			"name": "share",
			"proc_type": "shared",
			"processors": "0.5",
			"storage_config": [
			{
			"count": "1",
			"mount": "/sapmnt",
			"name": "sapmnt",
			"size": "300",
			"tier": "tier3"
			},
			{
			"count": "1",
			"mount": "/usr/trans",
			"name": "trans",
			"size": "50",
			"tier": "tier3"
			}
			]
				no

powervs_zone	IBM Cloud data center location corresponding to the location used in 'Power Virtual Server with VPC landing zone' pre-requisite deployment.	string	n/a	yes
prefix	Unique prefix for resources to be created (e.g., SAP system name). Max length must be less than or equal to 6.	string	n/a	yes
prerequisite_workspace_id	IBM Cloud Schematics workspace ID of an existing 'Power Virtual Server with VPC landing zone' catalog solution.	string	n/a	yes
sap_domain	SAP network domain name.	string	"sap.com"	no
sap_hana_master_password	SAP HANA master password.	string	n/a	yes

sap_hana_vars	SAP HANA SID and instance number.	object({
		sap_hana_install_sid = string
		sap_hana_install_number = string
		})
			{
			"sap_hana_install_number": "02",
			"sap_hana_install_sid": "HDB"
			}
				no

sap_solution	SAP Solution to be installed on Power Virtual Server.	string	n/a	yes

sap_solution_vars	SAP SID, ASCS and PAS instance numbers.	object({
		sap_swpm_sid = string
		sap_swpm_ascs_instance_nr = string
		sap_swpm_pas_instance_nr = string
		})
			{
			"sap_swpm_ascs_instance_nr": "00",
			"sap_swpm_pas_instance_nr": "01",
			"sap_swpm_sid": "S4H"
			}
				no

sap_swpm_master_password	SAP SWPM master password.	string	n/a	yes

Name	Description	Type	Default	Req
ssh_private_key	Private SSH key (RSA format) used to login to IBM Power Virtual Server instances. µŸ Should match to uploaded public SSH key referenced by 'ssh_public_key' which was created previously. µŸ Entered data must be in heredoc strings format.[a] µŸ The key is not uploaded or stored.	string	n/a	yes

a. https://www.terraform.io/language/expressions/strings#heredoc-strings

Table 4-3

4.5 Outputs

Table 4-4 shows the output variables for the Power Virtual Server for SAP HANA - SAP Ready Power Virtual Server:

Table 4-4 Output variables for Power Virtual Server for SAP HANA - SAP Ready

Name	Description
access_host_or_ip	Public IP of Provided Bastion/JumpServer Host.
infrastructure_data	Power Virtual Server infrastructure details.
powervs_hana_instance_ips	All private IPS of HANA instance.
powervs_hana_instance_management_ip	Management IP of HANA Instance.
powervs_lpars_data	All private IPS of Power Virtual Server instances and Jump IP to access the host.
powervs_netweaver_instance_ips	All private IPS of NetWeaver instances.
powervs_netweaver_instance_management_ips	Management IPS of NetWeaver instances.
powervs_sharefs_instance_ips	Private IPs of the Share FS instance.

Table 4-5 shows the output variables for the Power Virtual Server for SAP HANA - SAP S/4HANA or SAP BW/4HANA:

Table 4-5 Output variables for Power Virtual Server for SAP HANA SAP S/4HANA or BW/4HANA

Name	Description
access_host_or_ip	Public IP of Provided Bastion/JumpServer Host.
infrastructure_data	Power Virtual Server infrastructure details.
powervs_hana_instance_ips	All private IPS of HANA instance.
powervs_hana_instance_management_ip	Management IP of HANA Instance.
powervs_lpars_data	All private IPS of Power Virtual Server instances and Jump IP to access the host.
powervs_netweaver_instance_ips	All private IPs of NetWeaver instance.

powervs_netweaver_instance_management_ip	Management IP of NetWeaver instance.
powervs_sharefs_instance_ips	Private IPs of the Share FS instance.
sap_hana_vars	SAP HANA system details.
sap_solution_vars	SAP NetWeaver system details.

5

Troubleshooting Deployable Architecture implementation

This chapter covers some of the issues you may run into while using deployable architecture. The following topics are discussed:

© Copyright IBM Corp. 2025.

5.1 Common Deployment Issues and Solutions

This section focuses on frequently encountered issues during the deployment of Power Virtual Server architectures and practical solutions to resolve them.

5.1.1 Incorrect Terraform or Infrastructure as Code (IaC) Templates:

Symptoms

Deployment fails due to missing or misconfigured parameters in Terraform or Ansible scripts.

Solutions

- Run Terraform validate to identify syntax errors before deployment.
- Use sample Terraform scripts from IBM documentation as a starting point.
- Debug specific errors using Terraform plan output.

5.1.2 Authentication Errors:

Symptoms

API calls fail due to invalid credentials or IAM role misconfigurations.

Solutions

- Verify API key setup in the IBM Cloud dashboard.
- Assign proper IAM roles with least-privilege access to resources.
- "Test authentication with IBM Cloud CLI commands.

5.1.3 Network Connectivity Failures:

Symptoms

Instances cannot communicate across subnets or VPCs.

Solutions:

- Validate routing tables and ensure proper configuration of security group rules.
- Use IBM Cloud Network Troubleshooter to diagnose connectivity issues.
- Verify CIDR block configurations to avoid IP conflicts.

5.2 Performance Bottlenecks and Optimizations

Guidelines for identifying and resolving performance issues that may arise post-deployment.

5.2.1 Slow Network Throughput:

Symptoms

Delays in data transfer or high latency between VPC instances.

Solutions

- Optimize inter-zone connectivity using IBM Transit Gateway.
- Diagnose network issues using tools like traceroute or ping.
- Allocate additional bandwidth if required.

5.2.2 High CPU/Memory Utilization:

Symptoms

Overloaded instances resulting in application slowdowns.

Solutions

Monitor utilization metrics with IBM Cloud Monitoring.

Adjust instance size (scale-up) or distribute the workload (scale-out).

Optimize application configurations for resource efficiency.

5.2.3 Storage Latency:

Symptoms

Slow read/write operations affecting application performance.

Solutions

Use high-performance storage options such as IBM FlashSystem.

Separate data and log volumes for workloads like SAP HANA.

Enable caching mechanisms where applicable.

5.3 Debugging High Availability and Disaster Recovery (HA/DR) Setups

Focuses on resolving issues related to HA/DR configurations in Power Virtual Server environments.

5.3.1 Replication Failures:

Symptoms

Primary and secondary instances are not synchronized.

Solutions

- Check SAP HANA or application-specific replication logs for errors.
- Verify network bandwidth and latency between the replication nodes.
- Reconfigure replication settings and validate roles (primary/secondary).

5.3.2 Disaster Recovery Testing Issues

Symptoms
Failures during planned DR drills or recovery scenarios.

Solutions
- Use IBM VM Recovery Manager to simulate DR events and analyze logs.
- Verify RPO (Recovery Point Objective) and RTO (Recovery Time Objective) targets.
- Ensure DR site configurations align with production workloads.

5.3.3 Cluster Failover Failures

Symptoms
HA cluster does not automatically failover during node failure.

Solutions
- Examine PowerHA or similar cluster logs for errors.
- Verify quorum node configurations and ensure all nodes are reachable.
- Test failover processes periodically to ensure smooth operations.

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 soft-copy only.

- *IBM Power Systems High Availability and Disaster Recovery Updates: Planning for a Multicloud Environment*, REDP-5663
- *IBM PowerHA SystemMirror for i: Preparation (Volume 1 of 4)*, SG24-8400
- *IBM Power Systems Virtual Server Guide for IBM i*, SG24-8513
- *IBM Storage Virtualize and VMware: Integrations, Implementation and Best Practices*, SG24-8549
- *IBM Cloud Pak for Data Version 4.5: A practical, hands-on guide with best practices, examples, use cases, and walk-throughs, SG24-8522*
- *SAP HANA on IBM Power Systems Architectural Summary, REDP-5569*
- *SAP HANA on IBM Power Systems Virtual Servers: Hybrid Cloud Solution, REDP-5693*
- *IBM Power S1014, S1022s, S1022, and S1024 Technical Overview and Introduction*, REDP-5675

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- IBM Support portal:

https://www.ibm.com/support/home/

- Power Systems Virtual Server FAQ:

https://cloud.ibm.com/docs/power-iaas?topic=power-iaas-power-iaas-faqs

- Power Systems Virtual Server getting started:

https://cloud.ibm.com/docs/power-iaas?topic=power-iaas-getting-started

- Power Systems Virtual Server network architecture diagrams:

https://cloud.ibm.com/docs/power-iaas?topic=power-iaas-network-architecture-diagrams

© Copyright IBM Corp. 2025.

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

Introduction to Deployable Architectures

SG24-8576-00

ISBN

IBM

(1.5" spine)
1.5"<-> 1.998"
789 <->1051 pages

Redbooks

Introduction to Deployable Architectures

SG24-8576-00

ISBN

IBM

(1.0" spine)
0.875"<->1.498"
460 <-> 788 pages

Redbooks

Power Virtual Server Deployable Architecture

SG24-8576-00

ISBN

IBM

(0.5" spine)
0.475"<->0.873"
250 <-> 459 pages

Redbooks

Power Virtual Server Deployable Architecture

(0.2"spine)
0.17"<->0.473"
90<->249 pages

(0.1"spine)
0.1"<->0.169"
53<->89 pages

Redbooks

Introduction to Deployable Architectures

SG24-8576-00

ISBN

IBM

(2.5" spine)
2.5"<->nnn.n"
1315<-> nnnn pages

Redbooks

Introduction to Deployable Architectures in IBM Power Virtual Server

SG24-8576-00

ISBN

IBM

(2.0" spine)
2.0" <-> 2.498"
1052 <-> 1314 pages

SG24-8576-00

ISBN

Printed in U.S.A.

Get connected

ibm.com/redbooks