Redbooks

ibm.com/redbooks

# Hybrid Cloud with On-premises Cloud on IBM Z and LinuxONE


Lydia Parziale

Richard Daniel

Pablo Paniagua

Ravi Kumar

James Roca

Anna Shugol

Cecilia Vales


Cloud

IBM Z

IBM

Redbooks

IBM Redbooks

# Hybrid Cloud with On-premises Cloud on IBM Z and LinuxONE

October 2022

SG24-8530-00

**Note:** Before using this information and the product it supports, read the information in "Notices" on page v.

**First Edition (October 2022)**

This edition applies to IBM z16, LinuxONE, Red Hat Open Shift, IBM Cloud Infrastructure Center

# Contents



# Notices

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

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

# Trademarks

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

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

CICS®
Db2®
GDPS®
IBM®
IBM Cloud®
IBM Cloud Pak®
IBM Cloud Satellite®
IBM Consulting™
IBM FlashSystem®
IBM Spectrum®
IBM Z®
Parallel Sysplex®
RACF®
Redbooks®
Redbooks (logo) ®
Satellite™
z Systems®
z/OS®
z/VM®

The following terms are trademarks of other companies:

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

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

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

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

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

# Preface

Enterprises invested millions of dollars and spent decades on developing their on-premises infrastructure. As technology advances, so has the need to leverage that infrastructure, yet still modernize the enterprises' software portfolio to use more contemporary infrastructure, tools, and languages.

Enterprises require a unified, flexible, cost-optimal hybrid cloud with portable microservices and a reliable and securable on-premises infrastructure for data-intensive AI and mission-critical workloads. A hybrid cloud can help enterprises integrate their public and private clouds with their on-premises infrastructure.

Red Hat OpenShift is the core element of hybrid cloud approach from IBM® with IBM Z® and IBM LinuxONE.

This IBM Redbooks publication explains the capabilities of an integrated on-premises cloud that is based on a Red Hat OpenShift environment and includes a full stack of virtualization options and infrastructure management options, such as:

- IBM Cloud® Infrastructure Center
- Red Hat OpenShift Container Platform
- IBM Cloud Paks®
- z/OS® Cloud Broker
- Red Hat OpenShift Data Foundation (ODF)
- IBM Spectrum® Scale

## Authors

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

**Lydia Parziale** is a Project Leader for the IBM Redbooks® team in Poughkeepsie, New York, with domestic and international experience in technology management, including software development, project leadership, and strategic planning. Her areas of expertise include business development and database management technologies. She is a PMI certified PMP and an IBM Certified IT Specialist with an MBA in Technology Management. She has been employed by IBM for over 30 years in various technology areas.

**Richard Daniel** is a Thought leader Architect in India. He has 22 years of experience in Mainframe Modernization, Cloud Platform Architecture, and Engineering. He holds a degree in Computer Science. His areas of expertise include zOS Systems Programming, Cloud Platform Engineering, and Modernization.

**Ravi Kumar** is a Thought Leader Level 3 Certified Technical Specialist at the IBM Consulting™ Organization, from Dallas, Texas. He holds Master of Business Administration (MBA) degree from the University of Nebraska-Lincoln, US. He has 30 years of experience in Information Technology consulting with various Fortune 500 companies. Currently, he is working with many customers in North America in leading their Hybrid Cloud Transformation journey. He has written extensively in the areas of Digital Transformation, IBM Db2®, Data Analytics, and Artificial Intelligence.

**Pablo Paniagua** is an IBM Z® Hybrid Cloud Client Technical Specialist in Spain. He has 5 years of experience in OpenShift and LinuxONE. He holds a degree in Telecommunications Engineering from Universidad Pontificia de Comillas. Before IBM, Pablo worked as an IT consultant on distributed systems. During his time at IBM, he has helped many customers with their Linux on Z environments and Hybrid Cloud Strategies.

**James Roca** is a Principal Z Client Architect and UK Chartered Engineer (CEng MIET) with a proven track record of leading and managing major strategy, architectural, and enterprise-scale digital transformation projects. He has more than 23 years of experience with IBM and more than 26 years of experience in the IT Industry. During his career with IBM, James has held several senior technical and architectural leadership positions, which have included working at the IBM Hursley UK Development Lab, the IBM China Development Lab, in a regional capacity with the IBM ASEAN Growth Market Unit, with IBM Australia and New Zealand, and with the IBM Corporation in the US.

**Anna Shugol** is a IBM Z Hybrid Cloud Solutions Engineer, working for a global team and based in Winchester, England. She has joined IBM in 2011. Anna holds a Computer Science degree from Moscow State Technical University n.a. Bauman. Her areas of interest include large systems performance, containers on Z, and digital assets.

**Cecilia Vales** is a Mainframe consultant who is based in Spain. She has experience in zSystems hardware and zSystems operating systems field. She holds a degree in industrial electronics and automation engineering from Universidad Carlos III de Madrid. Her areas of expertise include LinuxONE implementation and virtual machine administration.

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

Robert Haimowitz
**IBM Redbooks, Poughkeepsie Center**

Patrik Hysky
**IBM Systems Technical Sales Services, Austin**

Tom Ambrosio, Bill Lamastro,
**IBM CPO**

Seeman Mannan, Saeed Majidi, David Stark
**IBM**

## Now you can become a published author, too!

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

Find out more about the residency program, browse the residency index, and apply online at:

**ibm.com**/redbooks/residencies.html

## Comments welcome

Your comments are important to us!

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

- Use the online **Contact us** review Redbooks form found at:

  **ibm.com**/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, IBM Redbooks
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

## Stay connected to IBM Redbooks

- Find us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

# Introduction

Applications are a central part of any organization. Specific factors must be considered when selecting an infrastructure for the applications, such as security, availability, adherence to the corporate Service Level Agreements (SLAs), flexibility, scalability, integration with other systems, and data gravity.

The cloud is no longer just a destination, but an experience. A hybrid cloud infrastructure must accommodate all of those application requirements and peculiarities to provide the best user experience for the application consumers.

In this publication, we use a lightweight, open source, microservices-based application that is called `Voting ap`. This application is deployed in our hybrid cloud environment that is running on z Systems®. This lightweight application can be found at this GitHub web page.

In this IBM Redbooks publication, we demonstrate that the application uses the agility and availability of the cloud platform (Red Hat OpenShift), while inheriting all of the core strength capabilities (security, scalability, and performance) of z Systems.

This chapter describes the z Systems Hybrid Cloud Platform that we implemented in our IBM lab environment. We provide a brief overview of the technologies that support z Systems and LinuxONE and the business value of the use of z Systems for your hybrid cloud. We also provide a high-level perspective about how one solution works.

This chapter includes the following topics:


- “Hybrid cloud strategy overview” on page 2
- “Hybrid cloud platform” on page 2
- “On-premises hybrid cloud on z Systems or LinuxONE” on page 3

## 1.1 Hybrid cloud strategy overview

Enterprises are transforming and looking for ways to modernize their applications to cater to new business workloads. A hybrid cloud is one of the important strategies that help the enterprise to decide to build, deploy, and manage applications on the cloud.

Some enterprise applications might be running on traditional data centers (referred to as *on-premises*). Other applications might be running on a public cloud, such as IBM Cloud, Amazon Web Services (AWS), Google Cloud Services, and Microsoft Azure, or a private cloud that is set up by the enterprise.

Depending on the business needs, a customer can choose their strategy for a hybrid cloud. With a hybrid cloud, the customer can retain control of their Systems of Records.

Enterprises gained experience by monitoring their businesses and understanding when they need to scale up or scale down their IT services or expand to new regions. Post-COVID-19, digital transformation is rapidly changing businesses around the world.

Enterprises are exploring new options for faster time to market. As the business must expand, a hybrid cloud strategy is helping to meet modernization goals. New business initiatives, such as the Internet of Things (IoT), blockchain, artificial intelligence (AI), and machine language (ML) are good reasons for adopting a hybrid cloud strategy.

Hybrid cloud strategies enabled enterprises to combine the best of private and public cloud services to optimize each workload and move workloads across public and hybrid clouds based on business requirements.

## 1.2 Hybrid cloud platform

Enterprises use hybrid cloud platforms to focus on the business strategies by integrating their IBM z Systems and LinuxONE platforms with their public cloud environment that is hosted by a public cloud service provider, such as IBM Cloud, AWS, Google Cloud Services, and Microsoft Azure. This combination enabled enterprises to combine the best of private and public cloud services to optimize each workload and move workloads across public and private clouds based on business requirements.

The hybrid cloud platform helped customers to use z Systems and LinuxONE with Red Hat OpenShift to support applications across environments.

The use of a hybrid cloud platform enables the following benefits:

- Digitally enhance older applications
- Establish secure and compliant environments.
- Build a scalable and resilient environment.
- Help customers to adopt new technology
- Optimize cost and resources

Figure 1-1 shows a high-level overview of a general architecture of a hybrid cloud that is built on z Systems or LinuxONE and uses the Red Hat OpenShift platform.

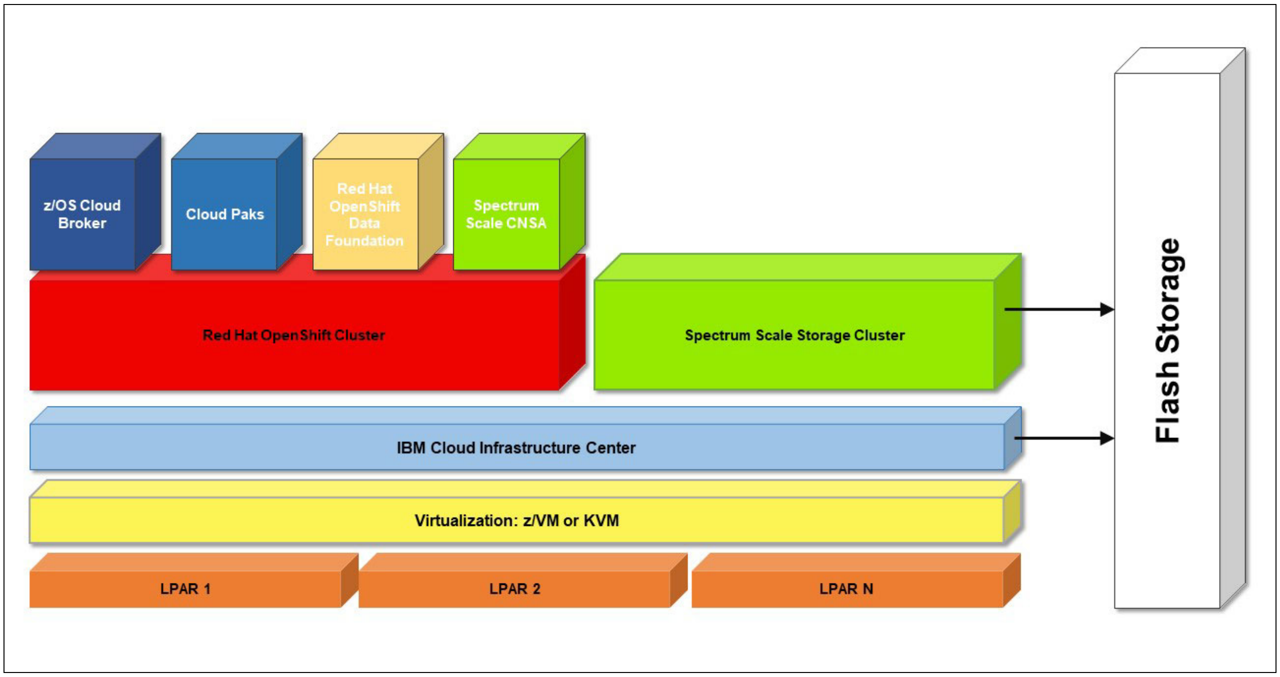



*Figure 1-1 General view of a hybrid cloud that is built on z Systems or LinuxONE with the Red Hat OpenShift platform*

## 1.3 On-premises hybrid cloud on z Systems or LinuxONE

Enterprises that are modernizing their applications often want to build on a hybrid cloud strategy. The combination of private cloud with z Systems or LinuxONE and public cloud with Red Hat OpenShift can provide the enterprise with the ability to cater to new business models.

The z Systems or LinuxONE with Red Hat OpenShift platform provides flexibility with superior reliability and scalability. It also builds confidence with secure data protection and privacy. By using this hybrid cloud strategy, the enterprise can build once and deploy anywhere. This strategy can also help to optimize IT to accelerate digital transformation and modernize applications to increase agility.

Hybrid cloud with Red Hat OpenShift can help customers to achieve better service level agreements (SLAs) at lower cost and seamlessly integrate with z Systems or LinuxONE where developers have greater agility. This platform also supports modernization and extends mission-critical older assets incrementally while maintaining enterprise SLAs and keeping risks and costs low.

When compared to public or private clouds, hybrid cloud is more open and inclusive of many architectures. Therefore, a hybrid cloud approach can help customers become more efficient.

# 2

# Introduction to IBM Cloud Infrastructure Center

In this chapter, we introduce and provide an overview of private clouds and an introduction to the IBM Cloud Infrastructure Center. We provide insight into IBM Cloud Infrastructure Center planning and deploying options.

We also present an overview of the problems that our use cases solve and describe our environment that we used to solve these problems.

This chapter includes the following topics:

## 2.1 Introduction to private clouds

The private cloud plays an essential role in the hybrid cloud environment because it is one of two key components in the hybrid model, along with the public cloud. However, the differences between the public and private clouds make them distinct concepts.

### 2.1.1 Commonalities

Starting with the common part of both cloud models, the first distinct characteristic of cloud computing is the accessibility through the "as a service" offering style. An increasing number of services can be accessed in a modern cloud, including the following examples:

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)
- Artificial Intelligence as a Service (AIaaS)
- Desktop as a Service (DaaS)
- Information Technology as a Service (ITaaS)

The most common of these services are IaaS, PaaS, and SaaS. The goal of the cloud model is for any user to access these services on-demand by way of the internet from any device.

### 2.1.2 Differences

Until now, what was often described was the client side of the cloud; however, what differentiates most private clouds from public clouds is the back end.

Other significant differences are the pricing options; however, for the purposes of this IBM Redbooks publication, only technical matters are addressed.

In the private cloud, the back-end provider is the same enterprise as the final client that uses the technology. The components that constitute the cloud infrastructure are the network, hardware, storage, and virtualization.

In the same way, a private cloud is a single tenant; that is, an architecture in which a single instance of a software application and supporting infrastructure serves one customer because only the data of a single enterprise is stored (the same company manages the cloud from both the server and client sides). This process does not occur in a public cloud environment because servers are shared, the system is multi-tenant and the cloud provider often is a different enterprise from the cloud services client enterprise.

In the public cloud model, the back-end infrastructure is not owned; instead, it is rented and used as a service.

### 2.1.3 Infrastructure as a Service

One of the most common services that is accessed in the modern cloud is IaaS. It is also the lowest level of provider-managed services because only the operating system is provided.

From this point, the customer can deploy anything from middleware to applications[1]. In today's enterprise architectures, agility is essential, and infrastructure must adapt dynamically and quickly to the needs of the users and applications.

[1] For more information, see https://www.ibm.com/cloud/learn/iaas

With agile standards, public cloud IaaS consumptions increased vastly because IaaS is easy to provision, is flexible, and requires little management for the user. This is true because the public cloud vendor performs all the requested reorganizations out of the users sight, so it is transparent and fast to the client and does not involve extra management costs whenever they want to perform any infrastructure modifications.

The public cloud can still provide IasS. It also can be a private cloud concept. By using the IBM Cloud Infrastructure Center, which is a private cloud IasS provider, these services can be obtained in an agile and easy way as the public cloud providers, with the benefit that it is dedicated and self-managed.

Other benefits of the private cloud IaaS are the native integration with the rest of the architectural components, and traffic compatibility and security.

Private and public clouds feature many differences, especially on the back-end side. However, both models have their benefits. It is at this intersection that the hybrid cloud concept becomes important to get the customizability of the private cloud and the efficiency and security.

At the same time, you can save on maintenance costs and gain in quick scalability with the hybrid cloud. Speaking more specifically about IaaS, all of the benefits of the private cloud also are realized with IBM Cloud Infrastructure Center, including the integration in the full stack model. IBM Cloud Infrastructure Center also provides an agile, customizable, and easy to use environment for infrastructure provisioning.

## 2.2 IBM Cloud Infrastructure Center overview

IBM Cloud Infrastructure Center is a powerful infrastructure management tool because it offers advanced capabilities for on-premises cloud deployments. IBM Cloud Infrastructure Center can be used to create virtual machines (VMs) in environments that are based on IBM z/VM, Kernel-based Virtual Machines (KVM) on z Systems, and IBM LinuxONE platforms.

As of this writing, hybrid hypervisor environments that combine z/VM and KVM cannot be managed from the same IBM Cloud Infrastructure Center[2].

IBM Cloud Infrastructure Center can be accessed by way of API to the web; the portal constitutes a self-service platform for VM resource provisioning. Resources for deployment can be dynamically assigned and managed from the resource pool that is provided to IBM Cloud Infrastructure Center. In addition, this tool is easy to use, even for nonadministrator roles.

The IBM Cloud Infrastructure Center offering covers deployments and utilities for managing the VMs. Some of these capabilities include start, stop, suspend, or restart of the machine.

It also is possible to capture an image of the machine, delete the VM completely, or get the console output of the VMs, which can be useful to check their state.

[2] For more information, see https://www.ibm.com/docs/en/cic

Aside from the basic functions, IBM Cloud Infrastructure Center offers editing the machine, attaching new volumes or resizing the number of processors, memory or disk ephemeral, and swapping size.

Another feature of IBM Cloud Infrastructure Center is that it allows migration of VMs from one hypervisor host to another, which is useful for high availability if an emergency occurs, and maintenance. These migrations can be live; that is, the VM does not need to stop to be moved to another host.

VMs that were not created by the compute node of IBM Cloud Infrastructure Center also can be managed in z/VM.

In addition to provisioning VM instances into on-premises cloud easily through the self-service web portal, IBM Cloud Infrastructure Center offers integration into higher-level cloud automation and orchestration tools. This feature allows deployments to be automated and auto-configured from the resources pool. In this way, infrastructure provisioning can follow policy or standardize workflows.

Also, the integration is absolute with other on-premises clouds and OpenStack clouds because of the compatibility with OpenStack APIs, which allows multi-region clouds and the automation of deployments on them. For this reason, IBM Cloud Infrastructure Center plays a key role in the integration with other clouds and other forms of services, such as PaaS with IaaS.

IBM Cloud Infrastructure Center offers an easy-to-use IaaS self-service portal that simplifies deployments and maximizes resource use. It also is a keytool in the full stack model and hybrid multi-cloud integration.

### 2.2.1 Planning and deploying options

IBM Cloud Infrastructure Center offers deployments of IBM z/VM-based and KVM-based Linux VMs. Depending on the option that is chosen, the planning and architecture and the minimum hardware and software requirements varies.

Figure 2-1 on page 9 shows the components that are needed for a z/VM architecture.

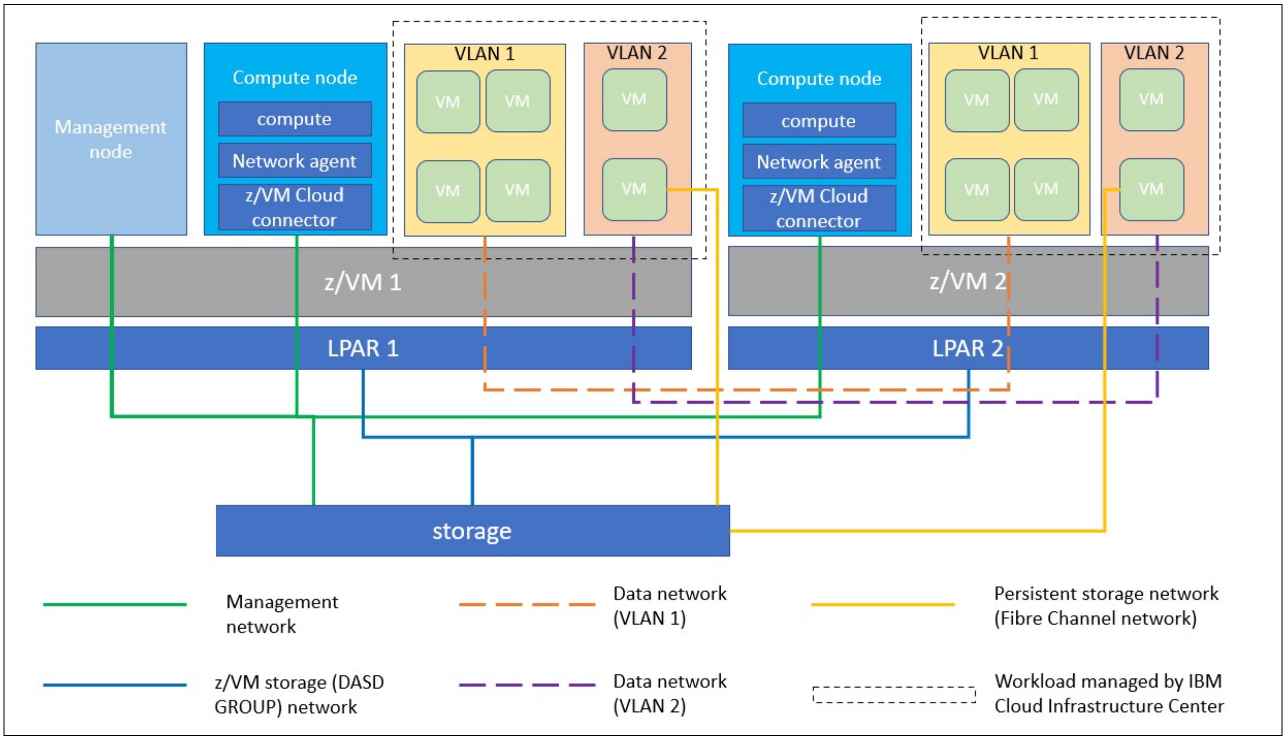



*Figure 2-1   Architecture of IBM Cloud Infrastructure Center on z/VM*

Only management node is required independently on the number of clouds and LPARs because one management node can control several compute nodes. However, a compute node is needed for every LPAR to be controlled by IBM Cloud Infrastructure Center.

Several VMs can be deployed inside each LPAR environment that is controlled by one compute node. These VMs can be deployed out of different images and be based on different distributions. In the same way, machines can belong to different VLAN data networks and different volumes can be assigned to them out of the available storage resources.

The main difference between KVM architecture that is shown in Figure 2-2 on page 10 and z/VM is the management node. In this case, for KVM, the management node is required to be in a different LPAR from the KVM hosts because the management node must not be virtualized inside one of the KVM host hypervisor layers. The rest of the architecture is equivalent, and deployments and management, which are similar to the web portal. However, installation and configuration differ.

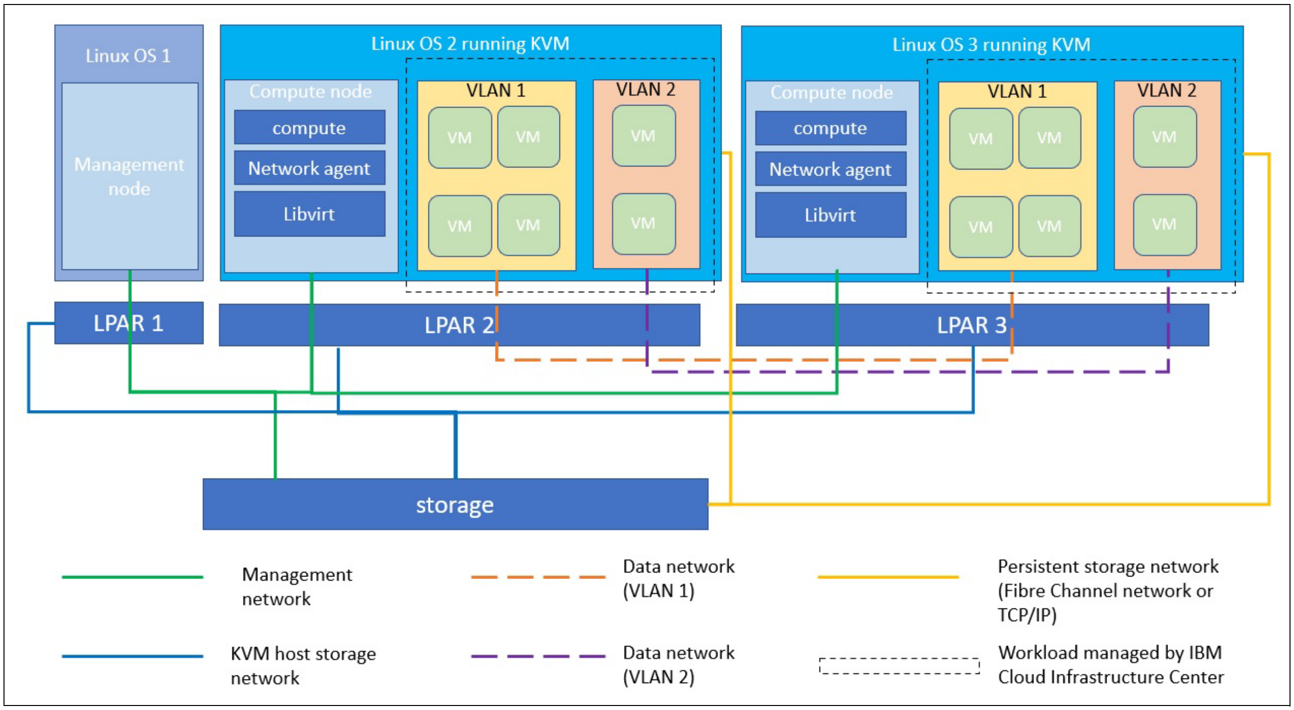



*Figure 2-2 Architecture of IBM Cloud Infrastructure Center on KVM*

Depending on the wanted architecture, planning for the installation and deployment can change. Some key points to consider when planning for the deployment are: networking, security, persistent storage and compute nodes, and hypervisors.

Also, IBM Cloud Infrastructure Center can work for a single cloud; however, if several clouds must be managed and the environment must be prepared to use the migration option, more than one compute node is necessary.

For our use example, only one management node is necessary because one node can manage multiple compute nodes. Based on the installation criteria, some configurations might differ from the steps that are provided in this publication.

For more information about the guidelines, see the IBM Documentation web page.

## 2.2.2 Our lab environment

In our lab environment, we used the IBM Cloud Infrastructure Center for our Linux VM deployments on z/VM. Because our environment consisted of only one compute cloud, it required only one LPAR.

For our architecture, we used one first-level z/VM hypervisor on IBM Z hardware (see Figure 2-3).

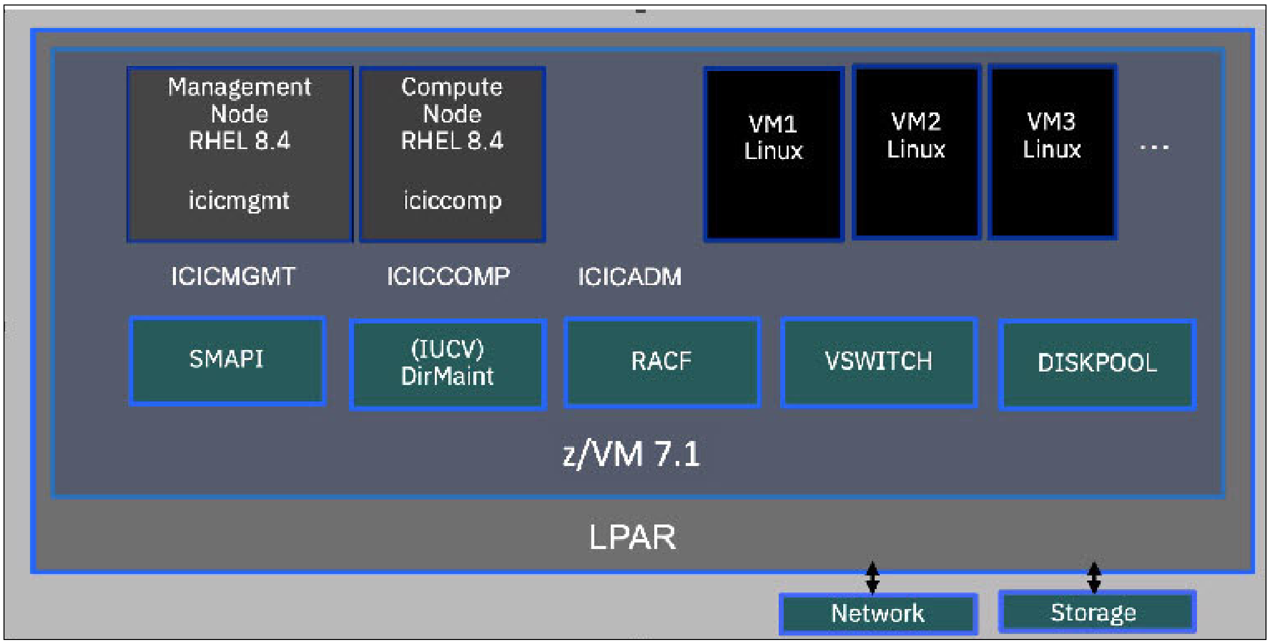



*Figure 2-3 IBM Redbooks lab environment*

The z/VM system is a member in an SSI cluster; however, this membership is not required. Our system used z/VM Version 7 Release 1.0, Service Level 2101. However, it is recommended to have z/VM updated to the latest level with the latest APARs[3].

The following tools must be configured and installed:

- z/VM SMAPI
- IBM RACF
- z/VM® DIRMAINT utilities

In our case, installed them with the latest APARs applied. If you have similar utilities from another supplier, contact your vendor to check compatibility.

[3] Check the latest service releases at http://www.vm.ibm.com/service/

For creating nodes, we created two Linux guests: one for management and one for compute nodes. Both machines include the operating system Red Hat Enterprise Linux 8.4 with the configuration details that are listed in Table 2-1. Minimum requirements are less than what we used because IBM Cloud Infrastructure Center hosts a Red Hat OpenShift Container Platform deployment. The number of IFLs we used is higher than minimum specifications.

*Table 2-1 Red Hat Enterprise Linux configuration details for z/VM*

| ICICMGMT (Management node) | ICICCOMP (Compute node) |
|---|---|
| ► Red Hat Enterprise Linux 8.4<br>► Memory: 16 GB<br>► Disk: 40 GB<br>► IFL: 1 | ► Red Hat Enterprise Linux 8.4<br>► Memory: 8 GB<br>► Disk: 80 GB<br>► IFL: 1 |

We created a virtual switch that is named ICICVS, which is required for the IBM Cloud Infrastructure Center installation on z/VM. For more information about the requirements for your specific installation, see this IBM Documentation web page.

## 2.2.3 Alternative environment: KVM

In the case of KVM, the infrastructure is similar to the infrastructure that is shown in Figure 2-2 on page 10. The difference is that we have only two LPARS because we have only one compute node. Therefore, we have two LPARs for KVM, the management node LPAR, and the compute node LPAR.

As for the z/VM scenario, both machines installed the Operating System Red Hat Enterprise Linux 8.4 and the characteristics that are listed specified in Table 2-2.

*Table 2-2 Red Hat Enterprise Linux configuration details for KVM*

| Management node LPAR | Compute node LPAR |
|---|---|
| ► Red Hat Enterprise Linux 8.4<br>► Memory: 16 GB<br>► Disk: 40 GB<br>► IFL: 1<br>► Boot partition: 1 GB | ► Red Hat Enterprise Linux 8.4<br>► Memory: 128 GB<br>► Disk: 600 GB<br>► IFL: 8<br>► Boot partition: 1 GB |

The minimum requirements are smaller than some of the numbers that were chosen. The reason is that IBM Cloud Infrastructure Center hosts a Red Hat OpenShift Container Platform deployment. The number of IFLs is then higher than the minimum specifications.

## 2.3 IBM Cloud Infrastructure Center installation for z/VM

We used the information that is available at this IBM Documentation web page as our roadmap for installation.

A step-by-step guide is available at this IBM Documentation web page.

Before starting, we reviewed Getting started with z/VM. It is important to have a clear understanding of the architecture (see Figure 2-1 on page 9) and we modified as shown in Figure 2-3 on page 11.

Also, before starting the installation, it is important to review the capabilities of the IBM Cloud Infrastructure Center and the features and support matrix.

You also need to ensure that you have the following z/VM prerequisite tools available:

- z/VM DIRMAINT
- z/VM local storage
- z/VM SMAPI
- z/VM vSwitch
- z/VM PROFILE

The prerequisite tools for z/VM are needed for the IBM Cloud Infrastructure Center to administer z/VM when creating or managing VMs. You also need to create the z/VM login IDs that are listed in Table 2-3.

*Table 2-3 Required z/VM login IDs*

| Log in ID | Function |
|---|---|
| ICICMGMT | Management node Linux guest |
| ICICCOMP | Compute node Linux guest |
| ICICADM | Used for installation |

z/VM login IDs ICICMGMT and ICICOMP match the Linux guests of the management and compute nodes, respectively. Log in ID ICICADM is used for installation.

### Preparing the hypervisor

When getting ready to set up the IBM Cloud Infrastructure Center in z/VM, determine whether the hypervisor belongs to a Single System Image (SSI) cluster. Because our z/VM is part of an SSI cluster and we are not managing several z/VMs from the same SSI cluster with the IBM Cloud Infrastructure Center, it is not necessary to perform any other steps.

If your setup differs, see the IBM Cloud Infrastructure Center Planning for the necessary steps for this process, which include planning and setting up for live guest relocation and shared local storage.

After you plan the architecture and complete the prerequisites, you must prepare the environment at the hypervisor level (in our case, z/VM).

#### *Creating the user IDs*

For the installation, the ICICADM login ID needs privilege classes ABCDEFG. You can check the privilege classes for this ID by running the **`query priv`** command.

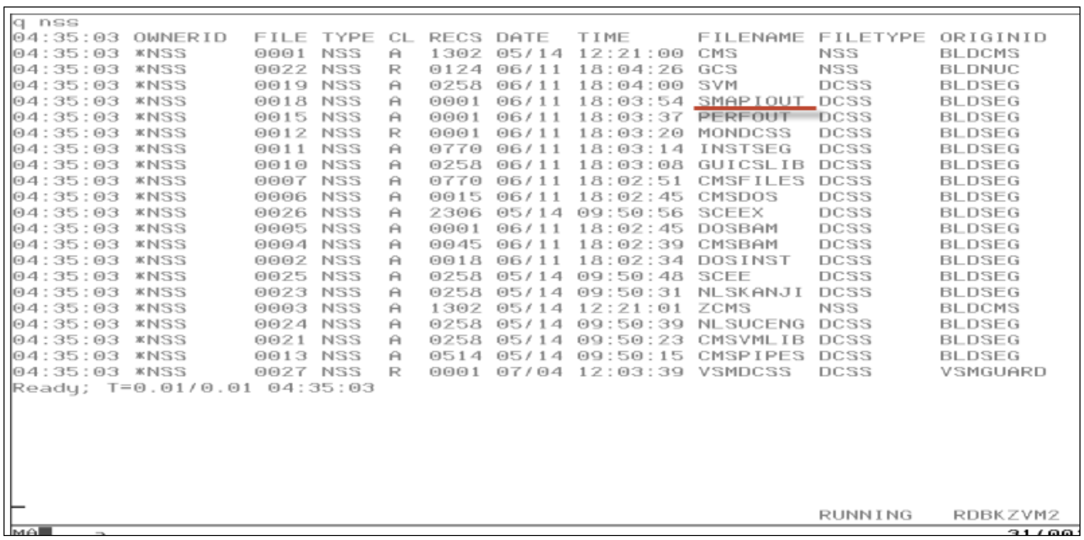
For more information about the user and functions for these IBM-defined privilege classes, see this IBM Documentation web page.

Verify that the privilege classes are correct by using the **query priv** command. Also, verify that all of the z/VM tools are installed and running correctly by using the **query names** command (look for the ID names of the z/VM software components, as listed in Table 2-4).

*Table 2-4 IDs and components they are attached to*

| Component | ID names |
|---|---|
| SMAPI | VSMREQIN, VSMEVSRV, VSMREQIU, VSMREQI6, VSMGUARD, VSMWORK1 |
| DirMaint | DIRMAINT, DATAMOVE |
| RACF® | RACFVM |

Verify that the Systems Management API (SMAPI) is enabled by using the **query nss** command. Figure 2-4 shows that all of our IBM z/VM tools are running correctly in our lab environment. You can also see in Figure 2-4 that SMAPI is running, as evidenced by the presence of the file name **SMAPIOUT**.



```
q nss
04:35:03 OWNERID   FILE TYPE CL RECS DATE  TIME     FILENAME FILETYPE ORIGINID
04:35:03 *NSS      0001 NSS  A  1302 05/14 12:21:00 CMS      NSS      BLDCMS
04:35:03 *NSS      0022 NSS  R  0124 06/11 18:04:26 GCS      NSS      BLDNUC
04:35:03 *NSS      0019 NSS  A  0258 06/11 18:04:00 SVM      DCSS     BLDSEG
04:35:03 *NSS      0018 NSS  A  0001 06/11 18:03:54 SMAPIOUT DCSS     BLDSEG
04:35:03 *NSS      0015 NSS  A  0001 06/11 18:03:37 PERFOUT  DCSS     BLDSEG
04:35:03 *NSS      0012 NSS  R  0001 06/11 18:03:20 MONDCSS  DCSS     BLDSEG
04:35:03 *NSS      0011 NSS  A  0770 06/11 18:03:14 INSTSEG  DCSS     BLDSEG
04:35:03 *NSS      0010 NSS  A  0258 06/11 18:03:08 GUICSLIB DCSS     BLDSEG
04:35:03 *NSS      0007 NSS  A  0770 06/11 18:02:51 CMSFILES DCSS     BLDSEG
04:35:03 *NSS      0006 NSS  A  0015 06/11 18:02:45 CMSDOS   DCSS     BLDSEG
04:35:03 *NSS      0026 NSS  A  2306 05/14 09:50:56 SCEEX    DCSS     BLDSEG
04:35:03 *NSS      0005 NSS  A  0001 06/11 18:02:45 DOSBAM   DCSS     BLDSEG
04:35:03 *NSS      0004 NSS  A  0045 06/11 18:02:39 CMSBAM   DCSS     BLDSEG
04:35:03 *NSS      0002 NSS  A  0018 06/11 18:02:34 DOSINST  DCSS     BLDSEG
04:35:03 *NSS      0025 NSS  A  0258 05/14 09:50:48 SCEE     DCSS     BLDSEG
04:35:03 *NSS      0023 NSS  A  0258 05/14 09:50:31 NLSKANJI DCSS     BLDSEG
04:35:03 *NSS      0003 NSS  A  1302 05/14 12:21:01 ZCMS     NSS      BLDCMS
04:35:03 *NSS      0024 NSS  A  0258 05/14 09:50:39 NLSUCENG DCSS     BLDSEG
04:35:03 *NSS      0021 NSS  A  0258 05/14 09:50:23 CMSVMLIB DCSS     BLDSEG
04:35:03 *NSS      0013 NSS  A  0514 05/14 09:50:15 CMSPIPES DCSS     BLDSEG
04:35:03 *NSS      0027 NSS  R  0001 07/04 12:03:39 VSMDCSS  DCSS     VSMGUARD
Ready; T=0.01/0.01 04:35:03

                                                   RUNNING      RDBKZVM2
```

*Figure 2-4 Output from the query nss command.*

To continue with the installation, start CMS and change from `CP READ` mode into `RUNNING` mode by running the **begin** command.

### *Configuring SMAPI*

You must configure the SMAPI tool because the compute node ID must be included in the list of SMAPI users. To configure this tool, list the SMAPI files by using the **sfsulist vmsys** command. This tool is not part of the standard z/VM installation; however, it can be downloaded from this web page if it is not installed.

Complete the following steps:

1. Highlight `VSMWORK1` and then, press PF11.
2. Highlight the `VMSYS:VSMWORK1` directory and then, press F11 to edit the `VSMWORK1 AUTHLIST` file.

3. Insert a line to add user ICICCOMP to the list and then, enter `file` on the command line to save the changes.

After including the user ICICCOMP to SMAPI, IBM RACF authorizations are required. Complete the following steps:

1. Run the **`RAC ALU ICICCOMP OPERATIONS`** command to provide access for the compute node to link available z/VM minidisks for provisioning VMs.
2. Allow the reader access to VSMWORK1 at SMAPI by using the following command:

   ```
   RAC PERMIT ICICCOMP CLASS(VMRDR) ID (VSMWORK1) ACCESS(UPDATE)
   ```

3. Restart SMAPI to load the new configuration by using the **`FORCE VSMGUARD`** command to stop SMAPI and the **`XAUTOLOG VSMGUARD`** command to restart SMAPI.

Now, SMAPI and RACF are configured for our installation of the IBM Cloud Infrastructure Center and for the operation of the tool.

### *Configuring IUCV*

Next, the Inter-User Communication Vehicle (IUCV) is configured. IUCV is required for communication between z/VM and the Compute Node (ICICCOMP) that is running as its Linux guest. DirMaint is used to set and check IUCV in a user ID profile.

DirMaint sets, checks, and controls this communication channel. For this reason, some modifications in the ICICCOMP user definition must be added. Complete the following steps:

1. Retrieve the user definition by using the **`dirm for ICICCOMP review`** command. The user's profile is sent to the reader list. Receive it from the reader by using the **`rl`** command.
2. Click **Clear** to list the reader list files.
3. Place your cursor on the **`ICICCOMP DIRECT`** line. Press PF11 to see the file (access the editor for the file). Verify that the line **`IUCV ANY`** is present. If it is not, you must add it to this user ID profile.

   The **`ANY`** option indicates that this z/VM guest VM can establish communication paths to any other z/VM guest VM. Extra security can be added by restricting the users that can exchange IUCV connections with the user ICICCOMP, instead of allowing **`ANY`**, which means all.

### *Creating and configuring a vSwitch*

A vSwitch is needed for the virtual guests to network the guests' systems. If a vSwitch is not defined on your system, complete the following steps to create one (a vSwitch requires three unused Open Systems Adapter (OSA) devices):

> **Note:** The **`QUERY OSA`** and **`QUERY OSA FREE`** commands report on the Open Systems Adapter (OSA) resources.

1. Create the vSwitch (add your own switch name and the three OSA device numbers by running the following command:

   **`DEFINE VSWITCH`** *switchname* **`RDEV`** *device numbers* (up to three) **`ETH VLAN AWARE NATIVE NONE`**

**Note:** CP automatically discovers which controllers are present and automatically balances the vSwitches across the available controllers. For the VLAN, you typically want to tell the vSwitch to be aware of VLAN tagging if it is used by setting VLAN AWARE, followed by NATIVE NONE so that it does not attempt to route packets that are untagged to VLAN 1. VLAN 1 can be thought of as a catch-all and it is considered bad practice to route packets that are untagged.

2. Verify that the vSwitch is correctly defined and working by using the **`q vswitch`** command with the name of your vSwitch.
3. Verify that the OSA devices that are assigned to the vSwitch are no longer free but instead are assigned to the vSwitch by using the **`q vswitch`** *switchname* command.
4. Make the vSwitch permanently available, including after an IPL, by adding it to the system configuration. To make the vSwitch definition permanently available, you must configure the disk pool.

   The disks that are assigned with this structure must be labeled and formatted before they are used. For this process, we used the **`cpfmtxa`** utility that is on `mdisk 551` from **`pmaint`**. Check that the disk devices are free. We attach each of the disk by using the **`attach`** command and the device number. We used the **`cpfmtxa`** command to format it.

   After the formatting process was complete, we detached the disks and then, queried those devices to ensure that they were correctly formatted and labeled.

   Figure 2-5 shows our example. We formatted the disks RDLXI0-RDLXI3 and RDLXIA-RDLXID to create two disk pools for IBM Cloud Infrastructure Center: one with smaller disks and another with larger disks.

```
 9062-9063
6:07:11 DASD 9062 CP SYSTEM RDLXI0    0
6:07:11 DASD 9063 CP SYSTEM RDLXI2    0
eady; T=0.01/0.01 06:07:11
 9162-9163
6:14:46 DASD 9162 CP SYSTEM RDLXI1    0
6:14:46 DASD 9163 CP SYSTEM RDLXI3    0
eady; T=0.01/0.01 06:14:46
 90EB-90EC
6:15:19 DASD 90EB CP SYSTEM RDLXIA    4
6:15:19 DASD 90EC CP SYSTEM RDLXIC    0
eady; T=0.01/0.01 06:15:19
 91EB-91EC
6:15:38 DASD 91EB CP SYSTEM RDLXIB    0
6:15:38 DASD 91EC CP SYSTEM RDLXID    0
eady; T=0.01/0.01 06:15:38
```

*Figure 2-5 Disks in the IBM Redbooks lab environment*

### *Creating disk pools*

The disk pools consist of groups of disks and are defined in the extent control file. Complete the following steps to create the disk pools:

1. Get extent control the file by running the **`dirm send extent control`** command.
2. Receive the file so you can make changes by using the **`receive / (replace`** command in the reader list.
3. Edit the file and add the new formatted disks to the REGIONS section with all of its data information. Our lab environment is shown in Figure 2-6.

```
:REGIONS.
*RegionId  VolSer   RegStart    RegEnd  Dev-Type  Comments
* USER VOLUMES
RDLXI0    RDLXI0       001       60100 3390-54
RDLXI1    RDLXI1       001       60100 3390-54
RDLXI2    RDLXI2       001       60100 3390-54
RDLXI3    RDLXI3       001       60100 3390-54
RDLXIA    RDLXIA       001       445199 3390-400
RDLXIB    RDLXIB       001       445199 3390-400
RDLXIC    RDLXIC       001       445199 3390-400
RDLXID    RDLXID       001       445199 3390-400
:END.
```

*Figure 2-6 IBM Redbooks lab regions environment*

4. Add the definition of the disk pools in the GROUPS section. The name of the disk pool and the new defined disk labels that are related to it must be included. For our example, we included two disk pools, as shown in Figure 2-7.

```
:GROUPS.
*GroupName RegionList
ICICSMLL RDLXI0 RDLXI1 RDLXI2 RDLXI3
ICICLRGE RDLXIA RDLXIB RDLXIC RDLXID
```

*Figure 2-7 GROUPS section*

5. Enter `file` in the command line to save the file.
6. Save the file into the system area by using the **`dirm file /`** command.
7. Reload the extent control file for activation by using the **`dirm rlde`** command.
8. Confirm that the disk pools were created correctly. The following commands use the names of our disk pools (substitute those names with the names that you created):

```
dirm free g= ICICSMLL
dirm free g= ICICLRG
```

The output of these commands is shown in Figure 2-8 and Figure 2-9 on page 18.

```
FREEXT G= ICICSMLL

  GROUP    REGION   VOLUME     START        SIZE        (END)     OWNER    ADDR
-------- -------- ------ ----------- ----------- ----------- -------- ----
ICICSMLL RDLXI0   RDLXI0           001       60100       60100 .FREE.   0000
ICICSMLL RDLXI1   RDLXI1           001       60100       60100 .FREE.   0000
ICICSMLL RDLXI2   RDLXI2           001       60100       60100 .FREE.   0000
ICICSMLL RDLXI3   RDLXI3           001       60100       60100 .FREE.   0000
```

*Figure 2-8 Output of dirm free g= ICICSMLL command*

```
FREEXT G= ICICLRGE

 GROUP    REGION   VOLUME    START      SIZE       (END)      OWNER    ADDR
-------- -------- ------ ---------- ---------- ---------- -------- ----
ICICLRGE RDLXIA   RDLXIA      14565       2914      17478 .FREE.   0000
ICICLRGE RDLXIA   RDLXIA      78648      69905     148552 .FREE.   0000
ICICLRGE RDLXIA   RDLXIA     250499      72818     323316 .FREE.   0000
ICICLRGE RDLXIA   RDLXIA     381572      63628     445199 .FREE.   0000
ICICLRGE RDLXIB   RDLXIB     116510     328690     445199 .FREE.   0000
ICICLRGE RDLXIC   RDLXIC        001     445199     445199 .FREE.   0000
ICICLRGE RDLXID   RDLXID        001     445199     445199 .FREE.   0000
```

*Figure 2-9 Output of dirm free g= ICICLRG command*

## Guest nodes

After the hypervisor is prepared, you are now ready to install the Linux guests for the management node and compute node, as shown in our lab environment in Figure 2-3 on page 11. We installed two Linux guests, ICICOMP and ICICMGMT, with the configuration details specified in Table 2-1 on page 12.

The installation process for the Linux guests depends on the operating system that you want to use. The installation process for IBM Cloud Infrastructure Center that we describe here is specific to Red Hat Enterprise Linux version 8.4.

Register your VMs on the portal and make sure that they both have available BaseOS and AppStream repositories. Networking must be configured as well, depending on your installation policies.

Complete the following steps to install the IBM Cloud Infrastructure Center on the management node Linux guest. For more information about detailed installation guide, see this IBM Documentation web page:

1. Create a file that is named **`config,properties`** in a new directory that is called `/etc/icic/`. In this file, add some configuration parameters for the creation of VMs by the IBM Cloud Infrastructure Center. In Example 2-1, we show the **`config.properties`** content of our lab environment machine.

*Example 2-1 Configuration parameters*

```
[root@icicmgmt ~]# cat /etc/icic/config.properties
[icic configs]
compute_instance_template=rdb%05x
compute_user_profile=ICICDFLT
default_admin_userid=MAINT
```

The following parameters are used:

– `compute_instance_template`: The name of the new instances that is created by the IBM Cloud Infrastructure Center.

– `compute_user_profile`: The z/VM login ID that is included in the USER directory of the VMs that were created.

– `default_admin_userd`: The privileged z/VM login ID that can log in to the VMs that are created. This process can be done with the LOGONBY function by adding the statement to the USER directory of the VMs by using this parameter.

For more information about parameters, see this IBM Documentation web page.

2. Configure networking. Use the **ifconfig** command to find the name of the LAN interface. As shown in Example 2-2, the **ifconfig** command returns **enc640**.

*Example 2-2 Output from ifconfig command*

```
[root@icicmgmt ~]# ifconfig
enc640: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
```

3. Set the `HOST_INTERFACE` value to the name of the LAN interface (that is, the value that was returned in the prior step), as shown in Example 2-3.

*Example 2-3 Setting the HOST_INTERFACE environment value*

```
[root@icicmgmt ~]# export HOST_INTERFACE=enc640
```

4. Ensure that the value of the environment value, LANG, is set to `en_US.UTF-8` at the file `/etc/locale.conf` (see Example 2-4).

*Example 2-4 Verifying the LANG environment value*

```
[root@icicmgmt ~]# grep LANG /etc/locale.conf
LANG="en_US.UTF-8"
```

5. Download the IBM Cloud Infrastructure Software pack if you have not done so.
6. Extract the contents of this file to the directory of your choice. Four different files are created, as shown in Example 2-5.

*Example 2-5 Decompressing the tar file*

```
[root@icicmgmt ~]# tar -xzvf IBM_Cloud_Infrastructure_Center_1.1.4.tar.gz
icic-install-s390x-rhel-1.1.4.0.tgz
icic-install-1.1.4.0.tgz.sig
icicpublickey
readme.txt
```

7. Decompress the `icic-install-s390x-rhel-1.1.4.0.tgz` file, as shown in Example 2-6. The contents of this file are extracted to the `Downloads/icic-1.1.4.0/` directory.

*Example 2-6 Decompressing the second tar file*

```
[root@icicmgmt ~]# tar -xzvf icic-install-s390x-rhel-1.1.4.0.tgz
```

8. Change to the working directory by using the **cd icic-1.1.4.0** command and then, start the installation by using the command that is shown in Example 2-7.

*Example 2-7 Running the install command*

```
[root@icicmgmt ~]# ./install -z -c
```

Where:

– option `z` specifies the z/VM hypervisor.

– option `c` automatically creates the necessary firewall rules. (This option is not needed if firewall rules are configured manually).

> **Note:** The installation files are the same for KVM and for z/VM, so it is required that you specify the hypervisor. In our lab environment, we specified `z` for z/VM.

For more information about options, see this IBM Documentation web page.

To check the status of the services after the installation, run the command that is shown in Example 2-8. Your output should look similar.

*Example 2-8 Check status of services*

```
[root@icicmgmt ~]# /opt/ibm/icic/bin/icic-services status

icic-bumblebee.service - IBM Cloud Infrastructure Center Bumblebee
   Active: active (running) since Tue 2022-06-28 12:22:41 CDT
ivp-validation-api.service - IVP API Server
   Active: active (running) since Tue 2022-06-28 12:22:41 CDT
clerk-api.service - clerk API Server
   Active: active (running) since Tue 2022-06-28 12:22:41 CDT
```

You should now be able to access the IBM Cloud Infrastructure Center portal (which is available on the management node) by using a web browser.

You are presented with a login window (see Figure 2-10) in which you log in by using the same user ID and password that you defined during the installation of the Linux management node. Extra users and projects can be defined and more security can be configured.

For more information about security options, see this IBM Documentation web page.

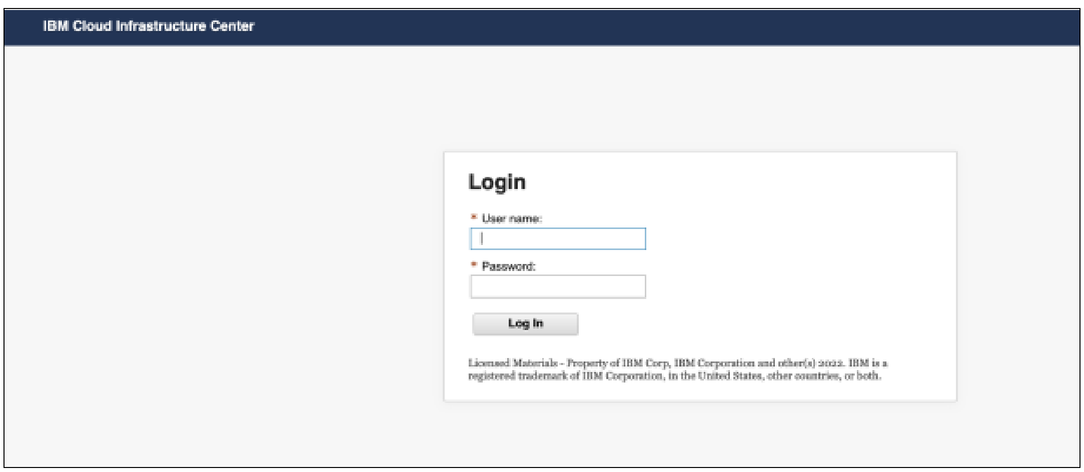



*Figure 2-10 Login window for web portal*

After successful login, you must include the compute node as a host. Complete the following steps to configure the hosts:

1. Select the **Hosts** menu item on the left side of the window. Click the **Hosts** tab and then, click **+Add host** from the top menu, as shown in Figure 2-11. A new window opens in which you enter the compute node information.

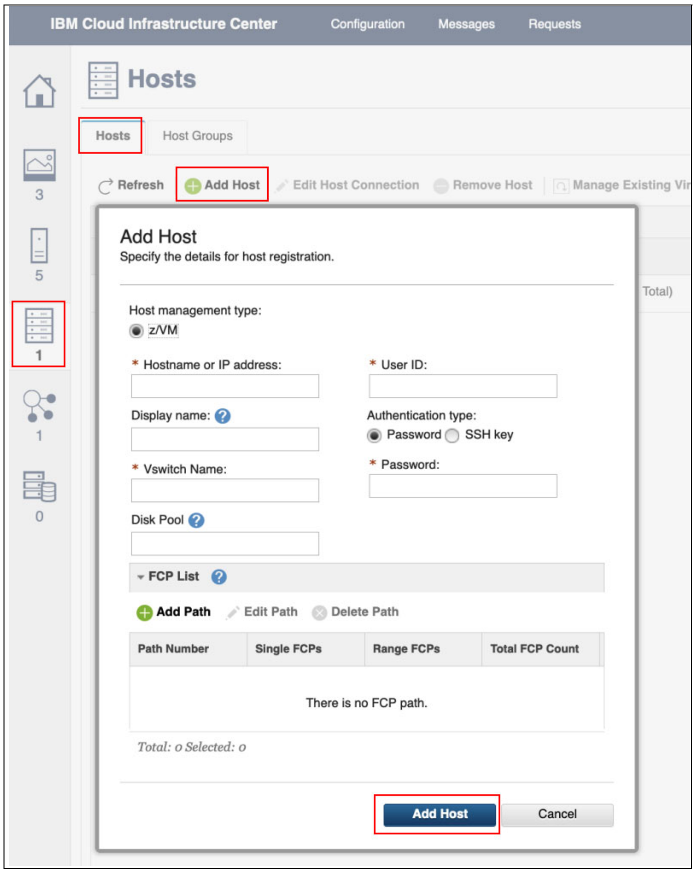



*Figure 2-11 Add Host window*

You must include the hostname or IP address of the compute node; that is, the name that the DNS can resolve. If you want to use a different name, you can change the name by adding one in the display name field.

The following information also is required:

– Vswitch name: The name of the vSwitch in z/VM.

– Disk Pool: The name of the disk pool in the following format:

`ECKD:<diskpool name>`

In our example: `ECKD:ICICLRGE`

– Log in information: The user ID and password for the compute node (in our example, root and its password.

2. Click **Add Host** to complete the process.

After adding your host information for the compute node, the next step is to obtain a master Linux image that is used to create VMs. IBM Cloud Infrastructure Center accept images in QCOW or RAW formats only; therefore, you must condition the image for its use in IBM Cloud Infrastructure Center if your image is not in an acceptable format.

For more information about creating an image that can be used by the IBM Cloud Infrastructure Center, see this IBM Documentation web page.

After you obtain an image in the correct format, uploading it to the IBM Cloud Infrastructure Center is a simple process.

Complete the following steps to create an image:

1. Select **Images** from the left side menu.
2. Click the **Images** tab.
3. Click **Create**. A window is displayed (see Figure 2-12).

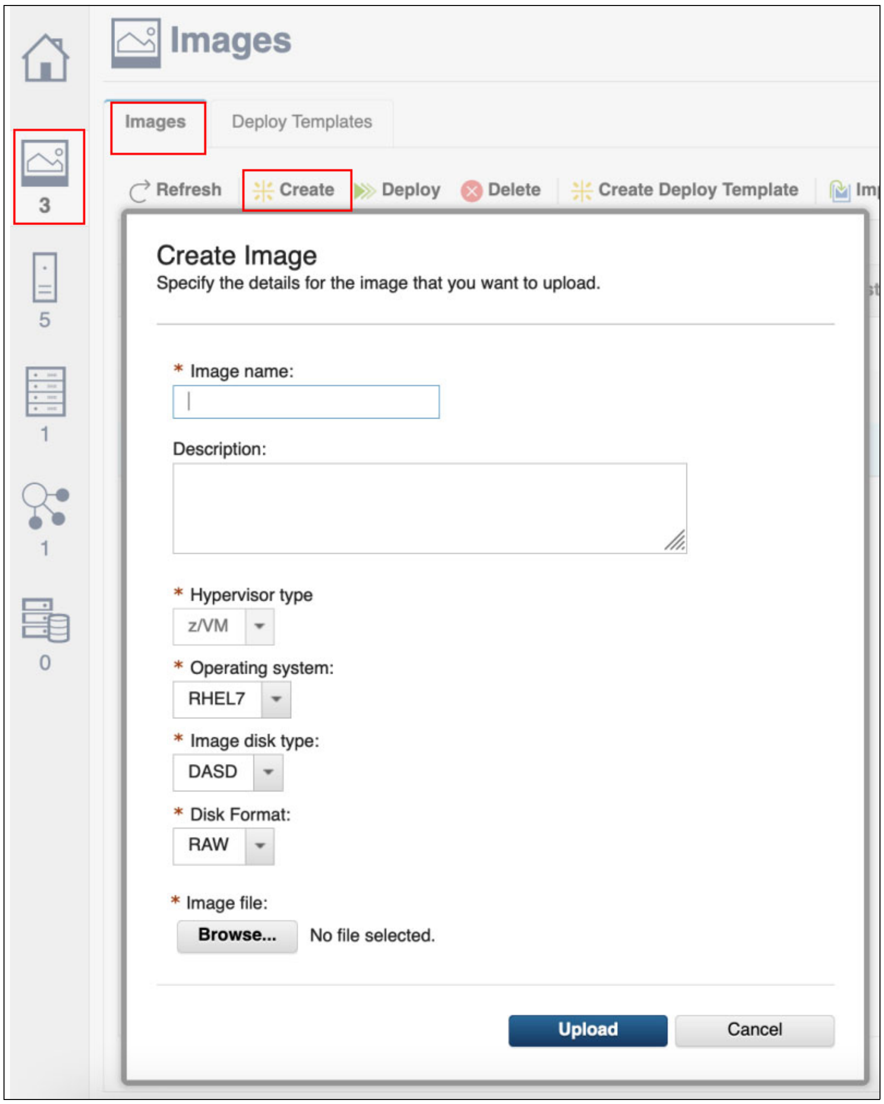



*Figure 2-12 Create Image window*

4. Complete the fields and select the image file for upload. Ensure that all fields are entered correctly (that is, hypervisor, operating system, disk type, and disk format). All fields must match the image that you are uploading.

As shown in Example 2-1 on page 18, we set the configuration parameters for the IBM Cloud Infrastructure Center. One of those parameters was `compute_user_profile=ICICDFLT`.

We now create a z/VM profile for ICICDFLT. The user profile for our lab environment is shown in Example 2-9.

*Example 2-9 ICICDFLT z/VM user profile*

```
PROFILE ICICDFLT
    COMMAND SET RUN ON
    COMMAND SP CONS * START
    MACHINE ESA
    OPTION APPL CHPIDV ONE
    CONSOLE 0009 3215 T
    SPOOL 000C 2540 READER *
    SPOOL 000D 2540 PUNCH A
    SPOOL 000E 1403 A
    LINK MAINT   0190 0190 RR
    LINK MAINT   019D 019D RR
    LINK MAINT   019E 019E RR
 *DVHOPT LNK0 LOG1 RCM1 SMS0 NPW1 LNGAMENG PWC20200803 CRCðZ
```

Before a VM is deployed in the IBM Cloud Infrastructure Center, you must set the provisioning network parameters in the IBM Cloud Infrastructure Center by completing the following steps:

1. Select the **Networks** window and then, click **Add Network**, as shown in Figure 2-13.

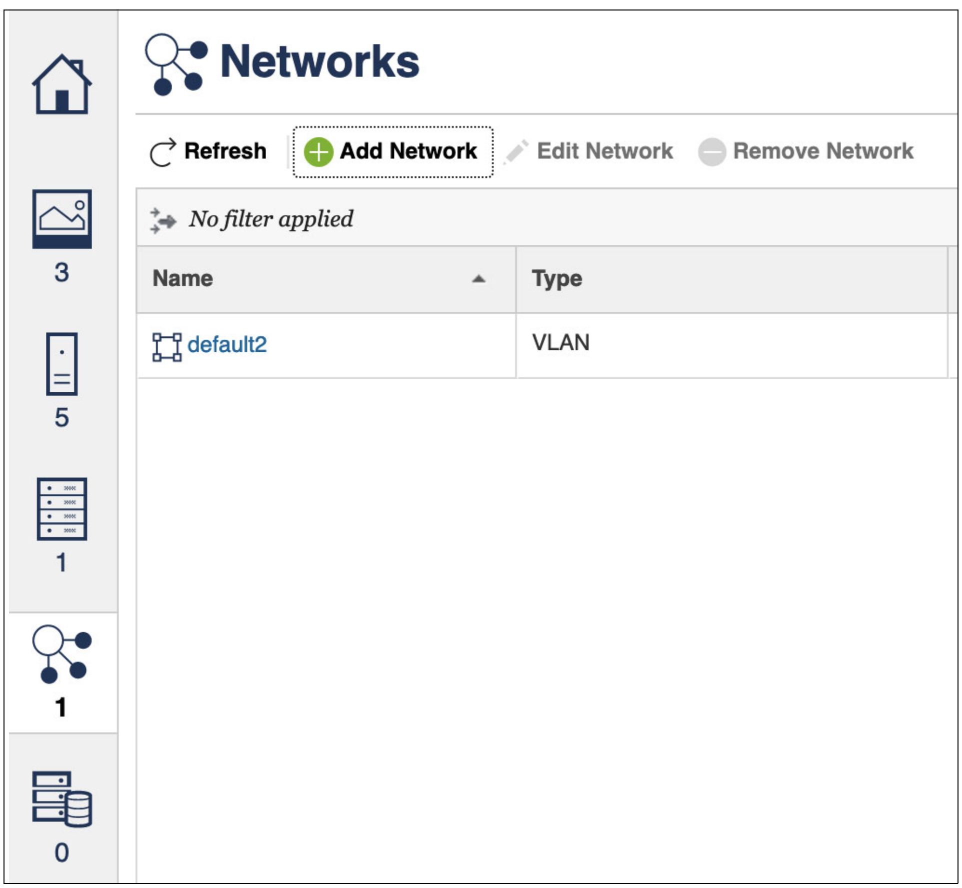



*Figure 2-13 Adding the network*

2. Enter the name, type of network, and type of virtualization. Parameters that are shown in Figure 2-14 are an example from our lab environment.

   Enter the subnet with a subnet mask, gateway IP address, and IP addresses that specify the primary DNS service. You also can enter the IP address of your secondary DNS service. You also can select the IP address range for provisioning VMs.

   In our environment, we created a VLAN with a VLAN ID of 8, and a virtualization type of z/VM vSwitch (see Figure 2-14).

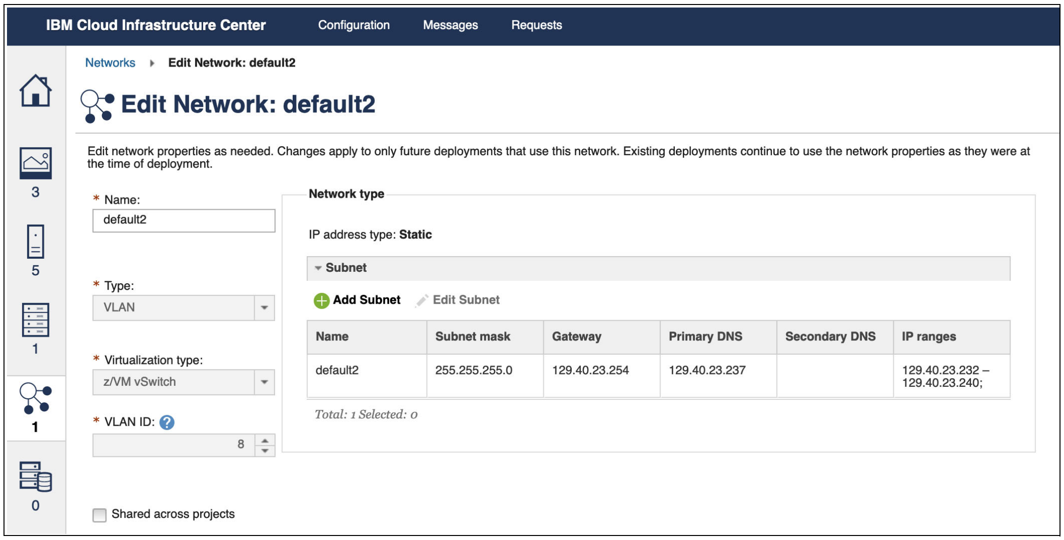



*Figure 2-14 Sample network parameters*

## 2.4 IBM Cloud Infrastructure Center installation for KVM

To install in a KVM environment, complete the following steps (for more information, see this IBM Documentation web page):

1. Run the **`dnf repolist`** command to verify whether you have the required BaseOS and AppStream repositories. Check your firewall settings to ensure that ports that are used by the IBM Cloud Infrastructure Center are not blocked.

   Also, check your web proxy because the IBM Cloud Infrastructure Center relies on internal web-based communication and needs an exception for its traffic. An example from our lab environment is shown in Example 2-10.

*Example 2-10 Exception for proxy*

```
export no_proxy="localhost,127.0.0.1,.example.com"
```

2. Create a configuration file that is called `config.properties` in the `/etc/icic` directory (you might need to create the `icic` directory). In this file, include the properties that customize your environment (similar to Example 2-1 on page 18 for z/VM).

   In this case, specify only the `compute_instance_template property` because the rest are specific to z/VM. The VM name template (`compute_instance_template property`) must be 8 characters or less or the VMs might not be created.

3. Unpack the installation components. This process can vary, depending on the installer package that was downloaded. For more information about this process, see this IBM Documentation web page.

4. Install the management node by using the `./install -k` command. You can decide whether to start the installation as silent, and an automatic firewall configuration, among other options. To check all of the available options, use the `./install -h` command.
5. After the command completes, the installation is complete. To confirm that services are active, use the `/opt/ibm/icic/bin/icic-services status` command.

   The logs of the installation can be found in the `/opt/ibm/icic/log` directory.

   If the installation fails, see this IBM Documentation web page to determine whether the problem you encountered is a known issue. For other issues, contact your IBM Customer Service Representative.

After the installation is complete, log in by using a web browser.

After successful login, you must include the compute node as a host, which must be configured. Complete the following steps to configure the host:

1. Select the **Hosts** menu item on the left side. Click the **Hosts** tab and then, click **+Add host** from the top menu, as shown in Figure 2-15. A new window open in which you enter the compute node information.

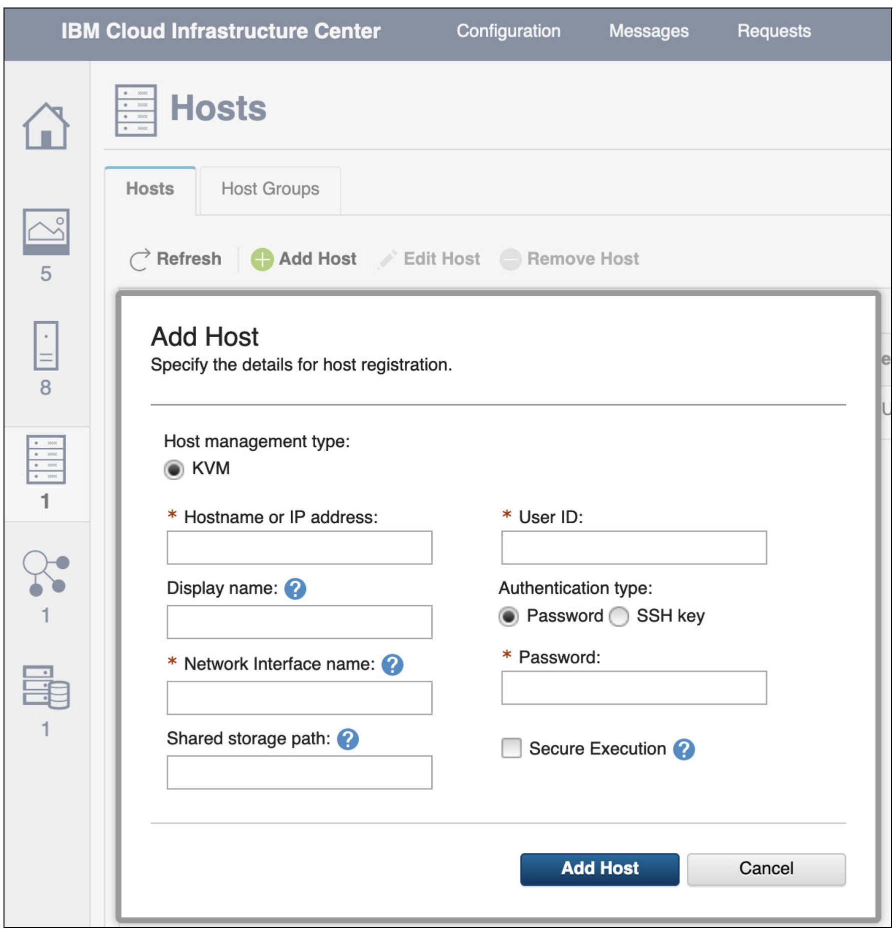



*Figure 2-15 Add Host*

You are required to include the Hostname or IP address of the compute node (that is, the name that the DNS can resolve). If you prefer to use a different name, you can change the name by adding one in the display name field.

Other information that must be included is the name of the network interface on the host that can be assigned to an Open vSwitch bridge. You also can add the storage path (it is not required now). Lastly, enter the log in information for the compute node (in our example, root and its password).

2. Click **Add Host** to complete the process.

To have IBM Cloud Infrastructure Center fully operational, add an image for later deployment of VMs. This process is similar to z/VM, as described in 2.2.3, “Alternative environment: KVM” on page 12.

Because the IBM Cloud Infrastructure Center detects the hypervisor, the only fields that are different in the form are automatically changed. To upload the image, complete the following steps.

1. Select **Images** from the left menu.
2. Click the **Images** tab.
3. Upload your file. The only valid formats for uploading are QCOW or RAW.

Next, the network must be configured. See 2.2.3, “Alternative environment: KVM” on page 12 because the steps are the same. After this process is complete, the IBM Cloud Infrastructure Center configuration and installation is completed for KVM.

## 2.5 Deployment from IBM Cloud Infrastructure Center

Simplified VM creation is a main advantage when IBM Cloud Infrastructure Center is used. This process is the same whether the hypervisor is z/VM or KVM because the interface also is the same for both.

Deployments from the IBM Cloud Infrastructure Center are based on images. Because you uploaded your images, complete the following steps:

1. Click the image that you want to deploy from in the Images tab and select Deploy as shown in Figure 2-16.

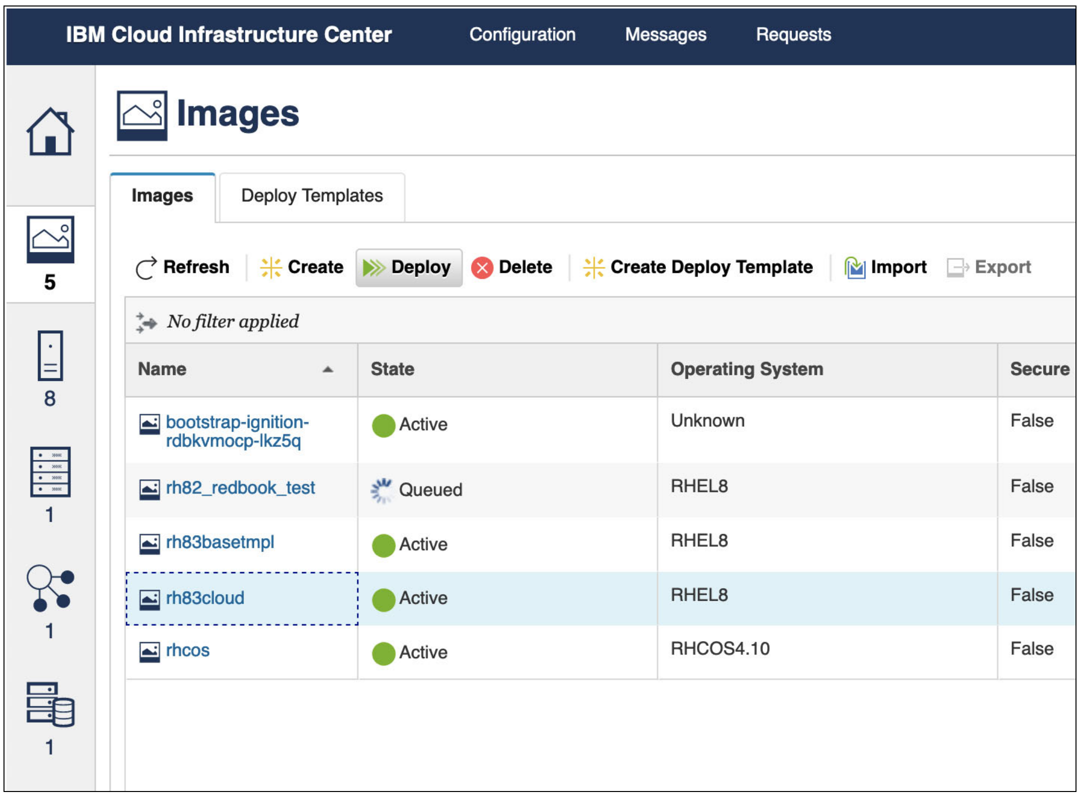



*Figure 2-16 Select an image to deploy*

2. Select the specifications for your new VM. The configuration includes specifying the name, description, number of instances, target of deployment, colocation rules, key pair, size specifications, and network as shown in Figure 2-17.

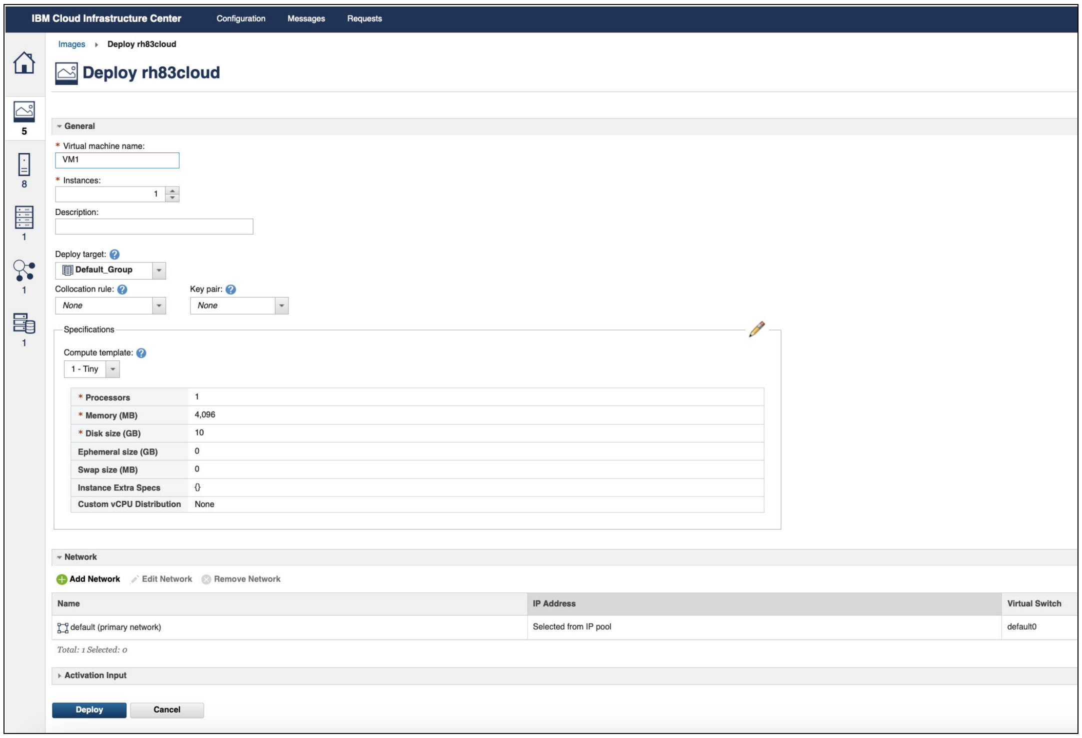



Figure 2-17 Deployment configuration

The target deployment parameter allows you to filter the network options, depending on the groups and projects the administrator defined.

Filtering the network options is useful because IBM Cloud Infrastructure Center allows users to directly create their own machines. In this way, each of the groups can have their own environment.

Colocation rules allow for the provision of VMs to be or not be on the same host as others and with affinity or anti-affinity rules.

Also, you can add a key pair to your machine and include SSH credentials. You can select the network for your machine and decide whether you want to specify an IP address manually or allow IBM Cloud Infrastructure Center to consider the IP range that was given to the network during the creation process.

By clicking **Deploy** after completing the form, your machine is fully deployed in one of the hosts that is defined in the IBM Cloud Infrastructure Center.

After the machine is deployed and active, you can perform several actions on the machine (see Figure 2-18), including the following examples:

- Start or stop the VM
- Suspend
- Restart
- Delete
- Capture a snapshot
- Resize the VM dynamically

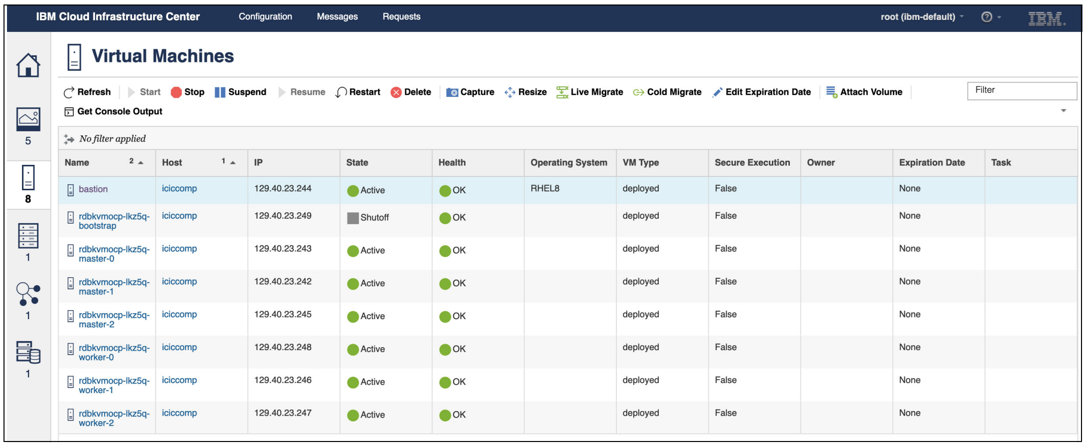



*Figure 2-18 Actions that can be performed on VMs*

To migrate from one host to another, more parameters must be configured. For more information, see this IBM Documentation web page.

A migration from one host to another can be live (without stopping) or cold (stopping the machine during the migration).

You also can attach more volumes to your VMs. With the storage devices defined in the storage tab, you can create volumes to assign to your VMs.

3

# Red Hat OpenShift overview

In this chapter, we provide an overview of Red Hat OpenShift and the benefits that are afforded when running on Z Systems and IBM LinuxONE.

We also provide and describe the Red Hat OpenShift reference architecture when it is deployed on Z Systems and IBM LinuxONE.

We also provide an overview of the Red Hat OpenShift Container Platform along with a description of provisioning infrastructure with the IBM Cloud Infrastructure Center.

Finally, instructions about automating the installation by using Red Hat Ansible are described, including the planning, prerequisites, and examples from our IBM Redbooks lab environment.

This chapter includes the following topics:



## 3.1 Overview

Red Hat OpenShift is an enterprise-ready container orchestration platform that helps organizations to build and deliver an open hybrid-cloud strategy.

Available for on-premises infrastructure, public cloud, and edge environments, Red Hat OpenShift provides the flexibility to deploy applications seamlessly across these disparate platforms, potentially from different vendors. As a result, a unified hybrid-cloud environment is created that is optimized for today's modern business demands.

For organizations with a significant on-premises infrastructure investment, including mission-critical workloads and data, Red Hat OpenShift can be instrumental in powering a strategic journey to the cloud by striking the correct balance between innovation, security, and speed to market. It also delivers cloud-native applications and digital transformation, while providing a foundation for modernizing older applications.

Red Hat OpenShift provides a consistent experience across all build, deploy, and run phases of an application lifecycle, regardless of the underlying infrastructure or cloud platform. Red Hat OpenShift delivers full-stack automated operations, self-service provisioning, and day-2 operations tools to accelerate cloud-native transformation initiatives.

Red Hat OpenShift includes the following features:

- An enterprise-grade Linux operating system
- Container run times and orchestration
- Networking capabilities
- Monitoring capabilities
- A container registry
- Authentication and authorization capabilities
- Serverless cloud computing model
- Service mesh
- Pipeline capabilities

At a high-level, Red Hat OpenShift delivers the following key benefits to development, operations, and business teams:

- Higher developer productivity
- Standardization and simplification
- Increased infrastructure efficiency
- Improved risk and compliance management
- Infrastructure cost efficiency
- Increased business acceleration

Red Hat OpenShift is available in various subscription models to meet customer demands and flexibility.

Red Hat OpenShift Container Platform is the self-managed, on-premises infrastructure edition of Red Hat OpenShift and is available on multiple hardware platforms, including x86, x86_64, ppc64, ppc64le, and s390x.

Red Hat OpenShift Container Platform Plus edition includes several other valuable components, including:

- Red Hat Advanced Cluster Management for Kubernetes
- Red Hat Advanced Cluster Security for Kubernetes
- Red Hat OpenShift Data Foundation Essentials
- Red Hat Quay

Cloud services fully managed editions of Red Hat OpenShift, include:

- Red Hat OpenShift Dedicated (on AWS and Google Cloud)
- Red Hat OpenShift Service on AWS
- Microsoft Azure Red Hat OpenShift
- Red Hat OpenShift on IBM Cloud

It also is possible to bring your own (BYO) Red Hat OpenShift Container Platform to the public cloud infrastructure (IaaS) vendor of your choice. This process includes installing and self-managing the deployment.

On Z Systems, Red Hat OpenShift also is available as part of the IBM zCX Foundation for Red Hat OpenShift (zCX for OpenShift) offering. This option gives z/OS environments the ability to deploy container-based applications (managed by Red Hat OpenShift) directly into z/OS address space.

IBM zCX Foundation for Red Hat OpenShift can be considered as the successor to the original IBM z/OS Container Extensions offering, the main difference being the switch from Docker Containers and Docker Swarm to OCI-/Docker Containers and Kubernetes, as found in Red Hat OpenShift.

IBM zCX Foundation for Red Hat OpenShift is *not* intended as a replacement for Red Hat OpenShift Container Platform on Linux on Z Systems/LinuxONE, which is the focus of this IBM Redbooks publication. Also, the entitlement of Red Hat OpenShift within IBM zCX Foundation for Red Hat OpenShift is nontransferable to Linux on Z Systems/LinuxONE.

In summary, by using Red Hat OpenShift at the core of your hybrid-cloud strategy, you gain a container orchestration platform with which you can deploy applications without any hardware dependencies, whether you are using on-premises infrastructure or the public cloud. This approach delivers the greatest degree of flexibility and negates infrastructure vendor lock-in.

## 3.1.1 Benefits of Red Hat OpenShift on Z Systems and IBM LinuxONE

The availability of Red Hat OpenShift on the Z Systems and IBM LinuxONE platforms is a major milestone for enterprise computing. It brings the immediate tangible benefits of containerization, cloud-native applications, and hybrid-cloud computing to the platform that delivers the highest levels of resiliency, scalability, and security in the industry.

### More secure and resilient foundation

Red Hat OpenShift on Z Systems and LinuxONE allows businesses to integrate and modernize their applications with a strong foundation that is built for security, resiliency, and availability.

Z Systems and LinuxONE prevent security threats and protect data across a hybrid cloud environment with certified multitenant workload isolation and a transparent, pervasive encryption with optimized performance. Z Systems and LinuxONE also protect the integrity and confidentiality of data with Crypto Express adapters (HSM) that meet strong security requirements of Federal Information Processing Standards (FIPS) 140-2 Level 4 and IBM Data Privacy Passports that protect data across private, public, and hybrid cloud.

Z Systems and LinuxONE hardware cryptographic support and key protection enables encryption everywhere for confidential computing. The unique combination of Red Hat OpenShift container security and the Z Systems cryptographic hardware creates a highly differentiated, security-rich solution.

### Flexibility and scalability

As organizations modernize applications to cloud-native architectures, the flexibility to manage and deploy the entire application portfolio across different infrastructures to scale is essential.

Red Hat OpenShift is a complete platform that complements features to build and deploy containerized software on any infrastructure. Together with Z Systems and LinuxONE, organizations can scale out to millions of containers on a single system for nondisruptive vertical and horizontal growth to accommodate increases of workloads on demand.

Teams can take advantage of high flexibility through dynamic resource sharing and reconfiguration. As a result, they also can continue to deliver excellent customer experiences with ultra low latency and large volume data serving and transaction processing.

### Efficiency for collocated workloads

Red Hat OpenShift on Z Systems and LinuxONE also optimizes latency, deployment, and cost through collocated containerized applications with data and applications. Cloud-native applications can be located close to workloads to improve throughput and reduce latency, which empowers organizations to integrate and modernize without disrupting services along their cloud-native journey.

Teams can centrally manage workloads by using a single platform that provides consistency across environments. Also, cloud developers can deploy IBM z/OS applications by using Red Hat OpenShift with no special Z skills required.

### Sustainability

Sustainability is now a strategic business imperative for many leading organizations. By deploying Red Hat OpenShift on Z Systems/LinuxONE, businesses can significantly reduce their energy consumption, and ultimately lower their carbon footprint, compared to running on commodity-based x86 hardware. This goal is achievable as Z Systems and LinuxONE delivers more performance per engine or core at a higher level of utilization, without exhibiting SLA degradation. This ability permits greater workload density and increased capacity.

## 3.1.2 Red Hat OpenShift on Z Systems and LinuxONE deployment reference architecture

Building a robust enterprise-grade hybrid cloud that delivers on the promise of agility and innovation, with the benefits of security, resiliency, and scalability, can be a complex challenge. With Red Hat OpenShift Container Platform that is deployed on Z Systems and LinuxONE, this objective can be reliably accomplished by implementing one of the IBM or Red Hat deployment patterns that already is validated.

The deployment patterns offer a repeatable archetype that can be automated for creating the deployment environment that best suits your needs.

### Production deployment

Production environments are where your applications and services are made available to your users or customers. Because a planned or unplanned outage of this environment can severely affect business, the deployment topology must include high availability by design.

Red Hat OpenShift Container Platform features a component-based architecture (see Figure 3-1) that consists of Control Planes and Compute Nodes. Each Control Plane includes a Kubernetes API server, etcd, Controller Manager Server, and other services, which are ultimately responsible for managing workloads on the Compute Nodes.

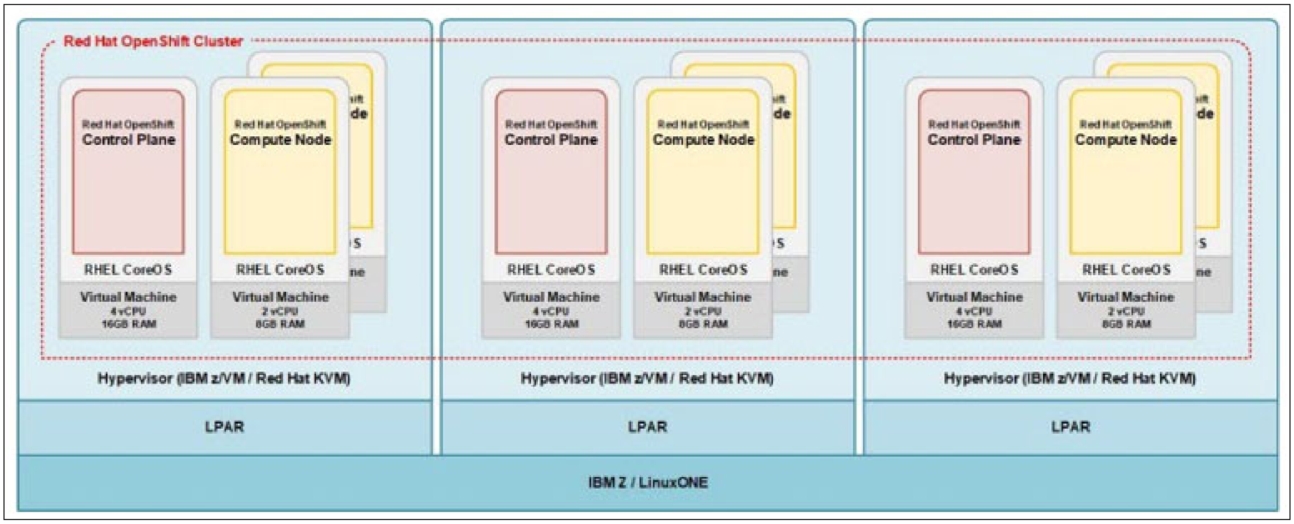



*Figure 3-1 Red Hat OpenShift Container Platform Production Deployment*

In a typical production environment, each of the Control Planes are installed on different virtual machines (VMs) and then, across multiple LPARs for redundancy.

Compute Node VMs also are distributed across multiple LPARs for redundancy, and can be further increased on each LPAR to address future scalability requirements, resources permitting.

To achieve 2+1 redundancy, three LPARs are configured into which IBM z/VM or Red Hat KVM hypervisors are installed to manage the require VMs.

Unlike commodity-based hardware, Z Systems and LinuxONE are built with enterprise redundancy by design. As such, the deployment can be made on a single Z Systems/LinuxONE server. The only exception to this configuration is when catastrophic site redundancy is considered. In this case, Z Systems/LinuxONE Capacity Backup (CBU) replaces lost capacity to a backup server if an unforeseen outage occurs.

IBM offers several options for site redundancy, including IBM Geographically Dispersed Parallel Sysplex® (IBM GDPS®).

## Proof of concept, technology, and experience deployment

A proof of concept, technology, and experience deployment (see Figure 3-2) is intended to demonstrate the feasibility of a solution. As such, the speed at which an environment can be provisioned often is more important than production environment redundancy.

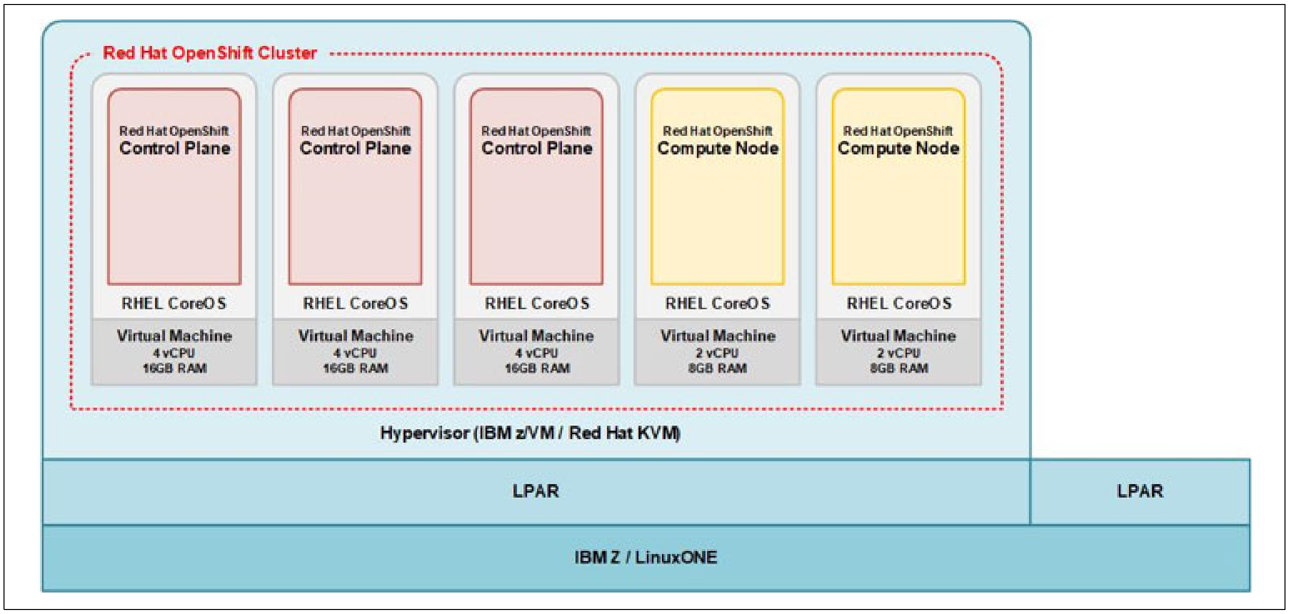



*Figure 3-2 Red Hat OpenShift Container Platform PoT / PoC Deployment*

This deployment pattern realizes the same overall operational architecture as the production environment deployment, but deploys the three mandatory Control Planes and two minimum Compute Nodes on a single Z Systems/LinuxONE LPAR. As such, the LPAR represents a single point of failure (SPOF) in the architecture.

## Minimal converged deployment

A new deployment model (see Figure 3-3) was introduced with Red Hat OpenShift 4.9 that converged Control Planes and Compute Node on to three VMs. This deployment topology is intended for proof of technology, development, or test purposes only.

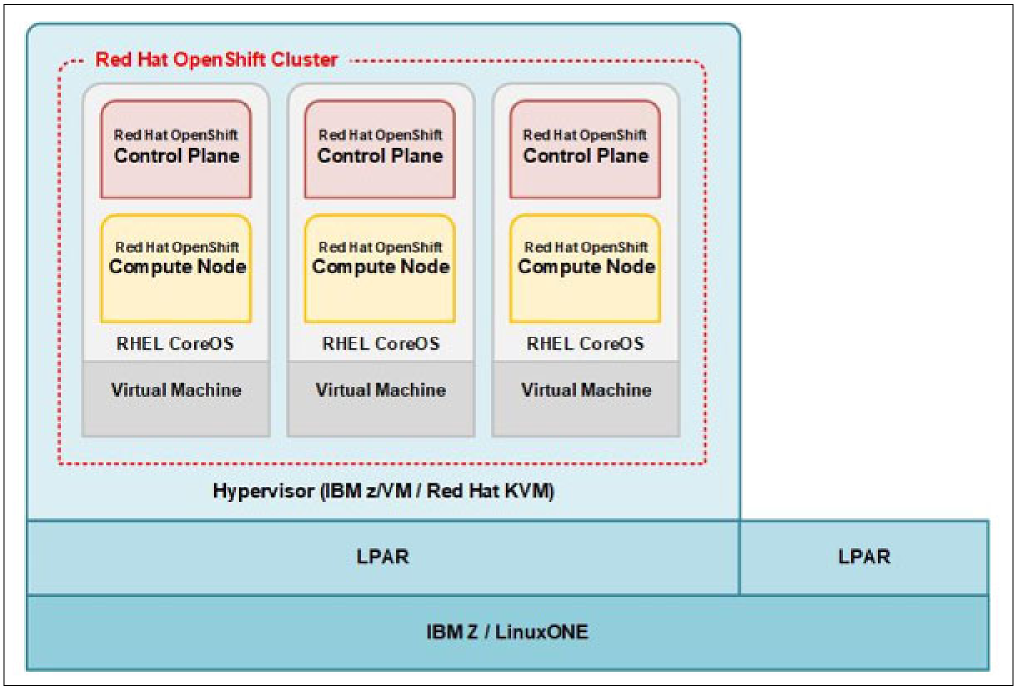



*Figure 3-3 Red Hat OpenShift Container Platform Converged Deployment*

## Network Architecture

Red Hat OpenShift Container Platform network requirements on Z Systems and LinuxONE, largely depend on your selected virtualization hypervisor. Although IBM z/VM and Red Hat Enterprise Linux KVM hypervisors support Open System Adapter (OSA) and RDMA Over Converged Ethernet (RoCE) adapters, a difference exists in terms of network interconnectivity.

For IBM z/VM deployments, VMs can be directly attached to multi-NIC OSA and RoCE adapters and onwards to an external switch or router. They also can be connected through a virtual switch (vSwitch) that is internal to z/VM, with external traffic then being routed externally. In addition, z/VM deployments can use the benefits of HiperSockets for internal connectivity between VMs.

In contrast, the RHEL KVM host must be configured to use bridged networking in libvirt or MacVTap to connect the network to the VMs.

All cluster components, control planes, and compute nodes must communicate with one another and each must resolve one another’s host name through DNS.

## Storage architecture

Red Hat OpenShift Container Platform includes specific storage requirements in terms of VM storage and container storage. For more information, see Chapter 4, “Storage options” on page 43.

## 3.2 Installing Red Hat OpenShift Container Platform

In general, Red Hat OpenShift Container Platform 4.x supports the following distinct cluster installation approaches:

- Installer-Provisioned Infrastructure (IPI)

  For this approach, infrastructure bootstrapping and provisioning are the responsibility of the *installer*. As such, the installer handles creating of all the networking, VMs, and operating systems that are required to support the cluster installation.

- User Provisioned Infrastructure (UPI)

  For this approach, the *user* is responsible for provisioning and managing the underlying infrastructure and resources, including all networking, compute, and storage. Although this approach allows for a greater degree of design freedom in terms of deployment topology, it can be more challenging to implement and maintain.

For Red Hat OpenShift Container Platform deployments on Z Systems and IBM LinuxONE, only UPI is supported. As such, you are responsible for provisioning and managing the following components:

- Load balancers
- Cluster networking
- DNS records
- All infrastructure that is related to the cluster, including the control plane and compute nodes
- Cluster infrastructure storage and container storage

In most cases, a production-ready cluster requires, at a minimum, six VMs and a temporary bootstrap VM during installation. A bastion host VM might also be required to host DNS, FTP, and HAProxy services.

IBM z/VM and Red Hat Enterprise Linux KVM on Z Systems and IBM LinuxONE are supported as hypervisors.

### 3.2.1 User Provisioned Infrastructure with IBM Cloud Infrastructure Center

Manually provisioning infrastructure is time consuming and error prone. But, what if a method existed to efficiently provision and manage the infrastructure that is required for establishing our Red Hat OpenShift Container Platform cluster as a service?

With IBM Cloud Infrastructure Center, you can do just that: provision and manage the lifecycle of Linux-based VMs that are deployed to the IBM z/VM and Red Hat KVM hypervisors.

IBM Cloud Infrastructure Center delivers a simplified Infrastructure-as-a-Service (IaaS) management experience with integration to cloud automation tools. It helps to abstract the complexities of the underlying hardware and software layers by providing a self-service portal for users.

For more information about IBM Cloud Infrastructure Center, see Chapter 2, “Introduction to IBM Cloud Infrastructure Center” on page 5.

## 3.2.2 Automating the installation with Red Hat Ansible

Although the IBM Cloud Infrastructure Center provides IaaS, one other element must be included to truly automate our Red Hat OpenShift Container Platform cluster installation.

By using Red Hat Ansible playbooks that are developed for the Red Hat OpenShift Container Platform installation on Z Systems and LinuxONE, the end-to-end setup and deployment of our cluster can be automated.

Ansible playbooks offer a repeatable, reusable, simplified configuration management and deployment scheme that is applicable to deploying complex applications.

At the time of this writing, the following publications are available for provisioning Red Hat OpenShift on Z Systems/LinuxONE. (In the case of deploying by way of IBM Cloud Infrastructure Center, select the second publication):

- Ansible-Automated Red Hat OpenShift Provisioning on KVM on Z Systems / LinuxONE
- Installing Red Hat OpenShift on the IBM Cloud Infrastructure Center via user-provisioned infrastructure (UPI)

## 3.2.3 Planning and prerequisites

The following planning and prerequisites must be met for installing Red Hat OpenShift Container Platform on Z Systems/LinuxONE by way of IBM Cloud Infrastructure Center:

- Red Hat OpenShift subscription

  A Red Hat OpenShift subscription is required for installing Red Hat OpenShift Container Platform on Z Systems/LinuxONE. A 60-day, self-supported trial is available at this Red Hat web page.

  After you signed up for a trial, you must create a Pull Secret that is then used during the installation to retrieve the Red Hat OpenShift software components.

- Internet connection

  A typical Red Hat OpenShift Container Platform installation requires access to the internet so that the Red Hat OpenShift software components can be downloaded by the bootstrap node. If you are behind a corporate firewall and cannot directly access the internet, you can use a restricted network installation. This process requires setting up and configuring at local registry that mirrors the contents of the Red Hat OpenShift Container Platform registry and contains the installation media.

- Systems requirements

  A Red Hat OpenShift Container Platform cluster features the specific minimum resource requirements that are listed in Table 3-1. The total number of vCPUs must equate to six IFLs or LinuxONE cores. When the IBM Cloud Infrastructure Center is used, each physical IFL or core can be divided 20 ways. Therefore, it is important to ensure that the six IFLs or LinuxONE cores requirement is met. The bootstrap VM is required only for installation purposes; therefore, it is required only temporarily.

*Table 3-1 Minimum resource requirements*

| Purpose | Operating system | vCPU | Memory | Storage |
|---|---|---|---|---|
| Bootstrap | RHCOS | 4 | 16 GB | 120 GB |
| Control Plane | RHCOS | 4 | 16 GB | 120 GB |
| Compute | RHCOS | 2 | 8 GB | 120 GB |

Other resources are required for the bastion host (and a separate HAProxy if configured); however, they are not considered as core components of the Red Hat OpenShift cluster.

- Bastion host

  A bastion host is considered a non-Red Hat OpenShift cluster component, but is required to host the DNS, HAProxy (load-balancer) and FTP servers when the infrastructure is unavailable.

  Traditionally, bastion hosts are commonly found in public-cloud deployments where they enable remote access by way of SSH or VPN to the Virtual Private Cloud (VPC) subnet where Red Hat OpenShift is deployed.

  In our lab environment, we also use the bastion host as the Linux Server, which is responsible for starting the Ansible playbooks. As such, it is necessary to install Red Hat Ansible on the bastion host.

  Red Hat Ansible Tower, which is part of the Red Hat Ansible Automation Platform, is not required.

- IBM Cloud Infrastructure Center dependencies

  IBM Cloud Infrastructure Center is required for our automated installation and configuration of Red Hat OpenShift Container Platform because it provides the provisioning and life-cycle management of the VMs. Sufficient compute resources, networking, and storage must meet the requirements that are listed in 3.2.3, “Planning and prerequisites” on page 39.

  For more information about selecting the underlying virtualization hypervisor (IBM z/VM or Red Hat Enterprise Linux KVM), see Chapter 2, “Introduction to IBM Cloud Infrastructure Center” on page 5.

### 3.2.4 Our lab environment installation

In our lab environment, we deployed a Red Hat OpenShift Container Platform cluster that matches the Proof of Concept, Technology, Experience Deployment deployment pattern that already is validated. It consisted of three control planes and three compute nodes that were deployed in a single LPAR, with a Red Hat Enterprise Linux host and Red Hat KVM hypervisor.

We first provisioned a bastion host VM that is running Red Hat Enterprise Linux 8 by way of the IBM Cloud Infrastructure Center dashboard. Contrary to the Ansible playbook instructions, we used the bastion host as the Linux Server that is responsible for starting the Ansible playbook scripts.

We also opted to use a DNS and HAProxy (load balancer) and manually updated their configurations after the installation. We disabled SELinux and the firmware on the bastion host.

After configuring the settings in the `inventory.yaml` file to suit our needs, we started the Ansible playbook scripts that are responsible for first preparing and then creating the cluster components. During these steps, we monitored the IBM Cloud Infrastructure Center dashboard for creating the respective VMs. It took some time for the control plane and compute node creation scripts to complete. During this time, the bootstrap node was busy downloading the Red Hat OpenShift software components.

We found that we needed to manually approve the cluster certificate signing requests (CSRs) after installation, during cluster bring-up. We manually approved the CSRS rather than wait approximately 10 minutes for each certificate to be approved automatically.

For more information about the Ansible playbook, see this Red Hat OpenShift web page.

After Red Hat OpenShift Container Platform is deployed and running, the administrator dashboard can be accessed in your browser, as shown in Figure 3-4.

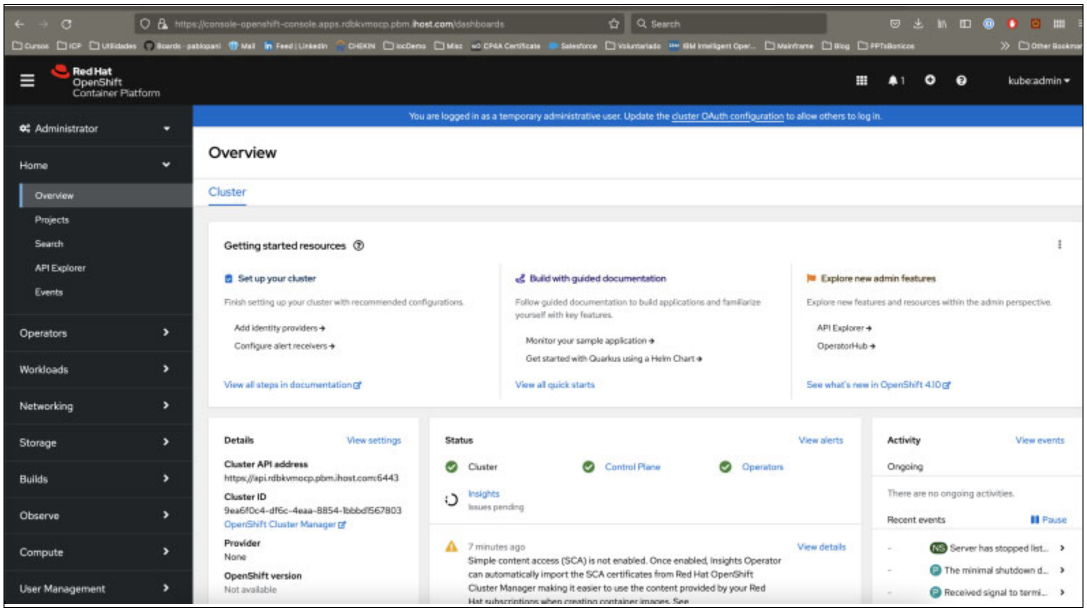


*Figure 3-4 Red Hat OpenShift Container Platform Administrator Dashboard*

## 3.3 Summary

Red Hat OpenShift Container Platform on Z Systems and LinuxONE enables organizations to embrace cloud-native technologies with the inherent benefits of the platform. It delivers the highest levels of resiliency, scalability, and security in the industry, which ultimately accelerates application modernization and digital transformation.

With IBM Cloud Infrastructure Center, infrastructure provisioning and life-cycle management are streamlined and greatly simplified.

Finally, by using Ansible playbooks, the deployment of Red Hat OpenShift Container Platform on Z Systems and LinuxONE is fully automated. This deployment fast-tracks the installation and configuration process, while minimizing the potential for user error.

4

# Storage options

This chapter discusses the different storage options that can be used with a hybrid cloud environment. The main idea is to cover storage in different parts of the storage stack.

In this chapter, we described at a high-level IBM Spectrum Scale, Red Hat OpenShift Data Foundation (ODF), IBM Spectrum Protect, IBM Spectrum Scale Container Native Storage Access (CNSA), and other useful tools.

This chapter includes the following topics:



## 4.1 Introduction to storage for the hybrid cloud

IBM Spectrum Fusion is the foundation for container-native applications that are running on Red Hat OpenShift. It provides enterprise-grade data storage and protection services.

Figure 4-1 shows how containers, cloud, and automation fit together in a hybrid cloud environment with IBM Spectrum Fusion Storage.

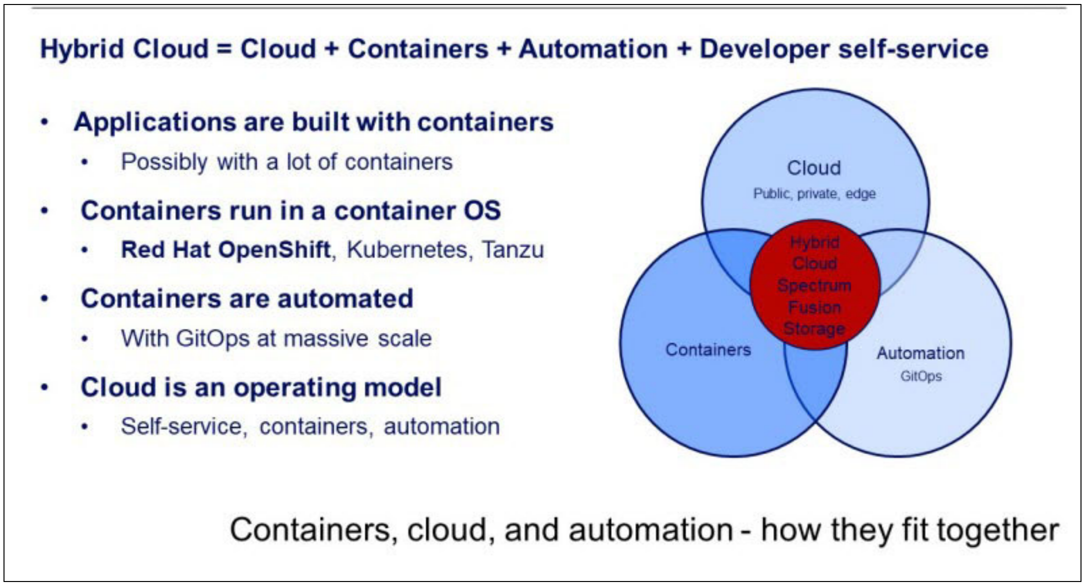



*Figure 4-1 Hybrid cloud IBM Spectrum Fusion storage*

The following salient features are available on Red Hat OpenShift Data Foundation and IBM Spectrum Fusion. These features offer a complete storage solution for all Red Hat OpenShift data needs in a hybrid cloud environment is possible:

- Red Hat OpenShift Data Foundation:
  - Immediately available simplicity of data services for Red Hat OpenShift
  - All major basic Kubernetes storage needs
  - Fully open source
- IBM Spectrum Fusion:
  - Builds on Red Hat OpenShift Data Foundation file, block, and object storage
  - Advanced data protection for Enterprise
  - Artificial Intelligence workloads with multiprotocol access
  - Global namespace sharing without copies
  - Metadata indexing, data discovery, and integrated data classification
  - Available as software or integrated hardware appliance

Figure 4-2 shows that IBM (Spectrum Fusion) and Red Hat (OpenShift Data Foundation) work together on common open-source projects, such as RamenDR and Red Hat OpenShift API for Data Protection (OADP).

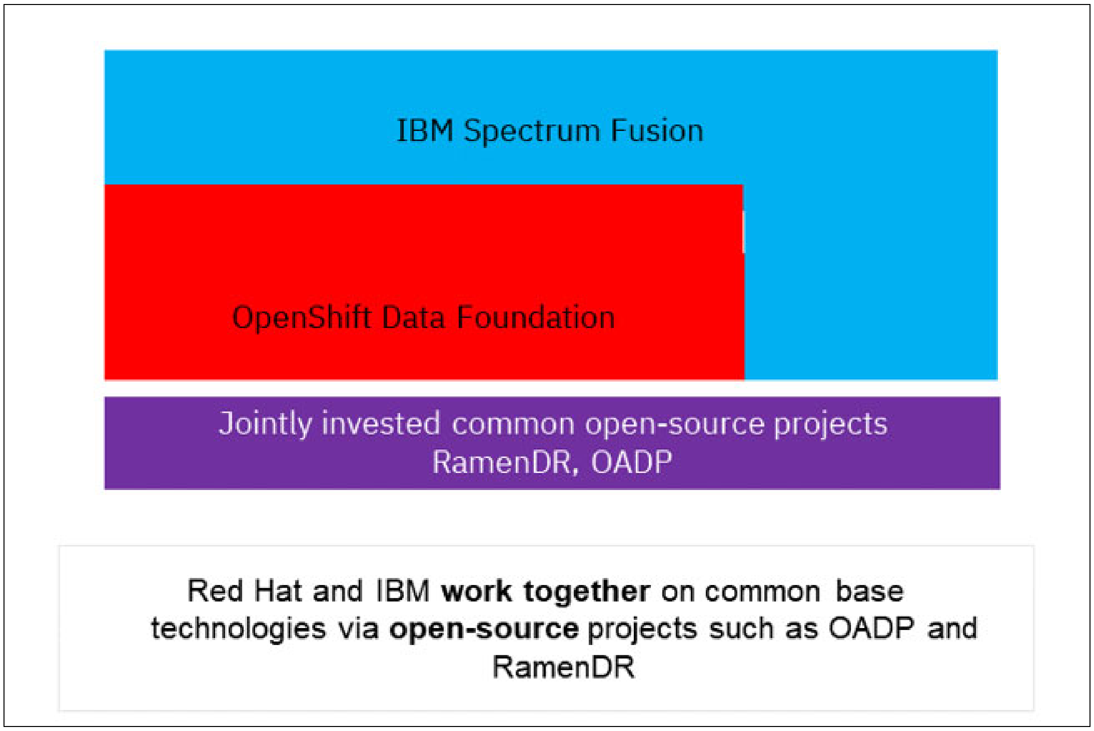



*Figure 4-2 Complete storage solution with IBM Fusion with Red Hat OpenShift Data Foundation*

OADP is a Red Hat OpenShift operator that provides Velero backup and restore APIs in the Red Hat OpenShift cluster.

IBM Spectrum Protect Plus uses OADP to back up and restore Red Hat OpenShift cluster resources (YAML files), internal images, and PV data.

By using OADP, IBM Spectrum Protect Plus can back up an entire container-based application, including all of its resources, metadata, and PVs. This feature allows fully functional applications to be cloned to other Red Hat OpenShift clusters for Disaster Recovery or development purposes.

Table 4-1 lists area wise features from Spectrum Fusion Hyper Converged Infrastructure (HCI).

*Table 4-1 Spectrum Fusion Hyper Converged Infrastructure Features*

| Area | Features |
|---|---|
| Compute | ► Bare-metal deployment of Red Hat OpenShift Container Platform to eliminate performance and cost overhead of hypervisors.<br>► Container-native hybrid cloud data platform for Red Hat OpenShift.<br>► Ability to run virtual machine (VM)-based workloads along side container-native workloads with Red Hat OpenShift Virtualization.<br>► The Red Hat OpenShift CoreOS virtualization can be used for supporting VMs on VMware.<br>► A single IBM Spectrum Fusion HCl cluster can go up to 20 nodes. |
| Storage | ► Integrated software-defined cloud-native storage.<br>► High performance, low latency NVMe local storage.<br>► Lower cost with transparent archive and data reduction.<br>► Foundation of Dynamic Deployment IBM Spectrum Scale Erasure Code Edition with Persistent Storage Container Storage Interface (CSI) plug-in.<br>► Flexibility Simultaneous support for multiple protocols: POSIX, NFS, SMB, HDFS, S3, CSI, and so on,<br>► Global access with Active File Management (replicate data caching global namespace to another location or Cloud).<br>► Global protection with cybersecure immutable capabilities.<br>► Container Persistent Storage runs on Red Hat OpenShift.<br>► ReadWriteOnce (RWO): The volume can be mounted as read-only by many nodes.<br>► ReadWriteMany (RWX): The volume can be mounted as read/write by many nodes.<br>► When a persistent volume claim (PVC) request specifies a Storage Class, the storage provider provisions the storage, creates a persistent volume (PV), and connects the PVC to the PV. |
| Backup | ► Integrated Spectrum Protect Plus backup and restore of application PVs and metadata to on-rack and off-rack back up targets.<br>► IBM Spectrum Protect Plus is BaaS Agent Helm and Operator deployment that is natively integrated to provide backups of the IBM Spectrum Fusion system, including the IBM Spectrum Fusion control plane and running customer applications on Red Hat OpenShift.<br>► VM-based management server with OVA deployment.<br>► Back up persistent data to remote vSnap and object storage with policy-driven backups.<br>► VMware or Red Hat OpenShift Snapshots backups and snapshots point-in-time recovery with application PV level backups. |
| Scalability | ► Simple to scale compute and storage nodes in the field.<br>► Best practice installation of Spectrum Scale Erasure Code Edition (ECE) is installed so that software that is deployed on Red Hat OpenShift easily can use the storage by using dynamic provisioning. The new node must have the same number of disks as the ones that are present. |
| High availability and encryption | ► Single point of contact for support of the complete solution.<br>► Spectrum Fusion underline Spectrum Scale provides data encryption both at rest and in flight.<br>► Enterprise Security that is centrally maintained for global data and access. |
| Disaster recovery | Active File Management support with replication of data to other data center or public cloud targets. |
| Services | ► Build Production ready from power-up in as little as 4 hours from Red Hat best practice operators.<br>► Integrated infrastructure management experience.<br>► Native IBM Cloud Satellite™ and Red Hat Advanced Cluster Management integration.<br>► Management UI for managing the HCI solution.<br>► Upgrade operator easily orchestrates ongoing firmware upgrades of firmware for hardware, Red Hat OpenShift, HCI Management Stack, IBM Spectrum Scale Erasure Code Edition (ECE) Storage. and IBM Spectrum Protect Plus Backups.<br>► Role-based access control (RBAC) objects support. |

## 4.1.1 Lab environment overview

The engrossment that was deployed in our lab environment is shown in Figure 4-3.

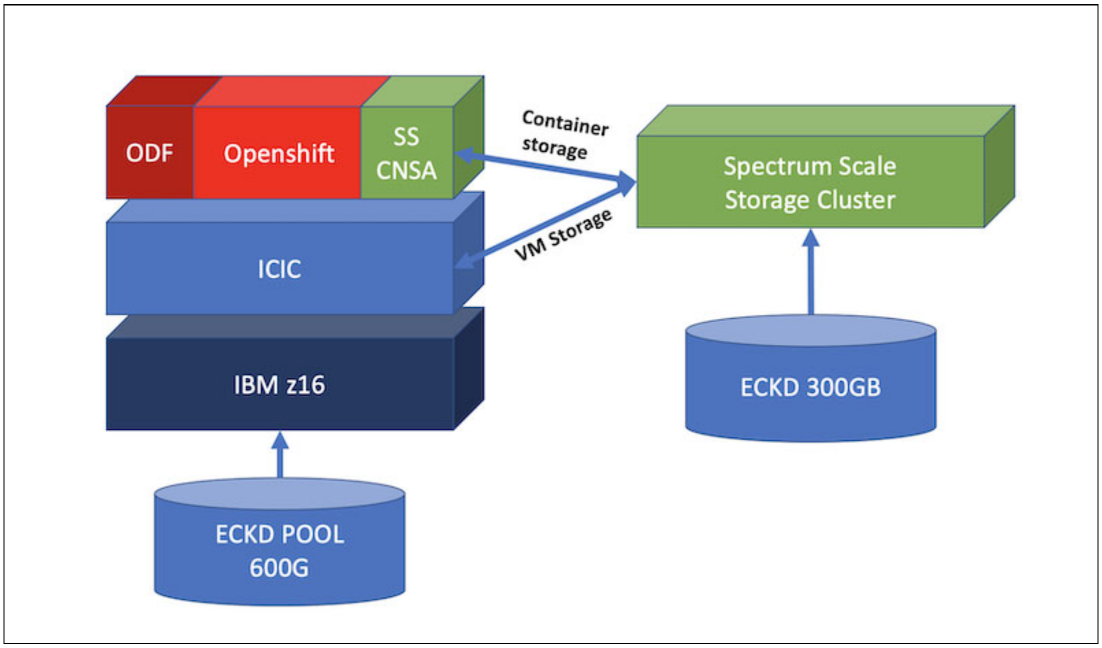



*Figure 4-3 General architecture of the storage deployed*

The following storage components are deployed:

- ECKD Disks: The disks that are used in our environment vary with the needs of each storage software. A total of 600 GB was provisioned for IBM Cloud Infrastructure Center to use and create VMs. Another 300 GB was allocated for IBM Spectrum Scale file systems.
- Spectrum Scale Storage Cluster: General Spectrum Scale cluster that services container (with CNSA) and VM storage. This cluster is used as the main way to provision storage in the lab environment.
- Spectrum Scale Container Native Storage Access: Deployed on top of Red Hat OpenShift to provision storage for containers.
- Red Hat OpenShift Data Foundation: Deployed on top of Red Hat OpenShift and uses internal storage of the cluster (also as an alternative for container storage).

### IBM Spectrum Scale in our environment

As part of the storage for the private cloud, an IBM Spectrum Scale storage cluster was created in the lab. This cluster consists of three nodes with one of them also acting as a GUI node. These nodes all have a 300 GB that is shared between nodes that are used by containers or VMs.

Figure 4-4 show a general architecture of the nodes and roles in the cluster.

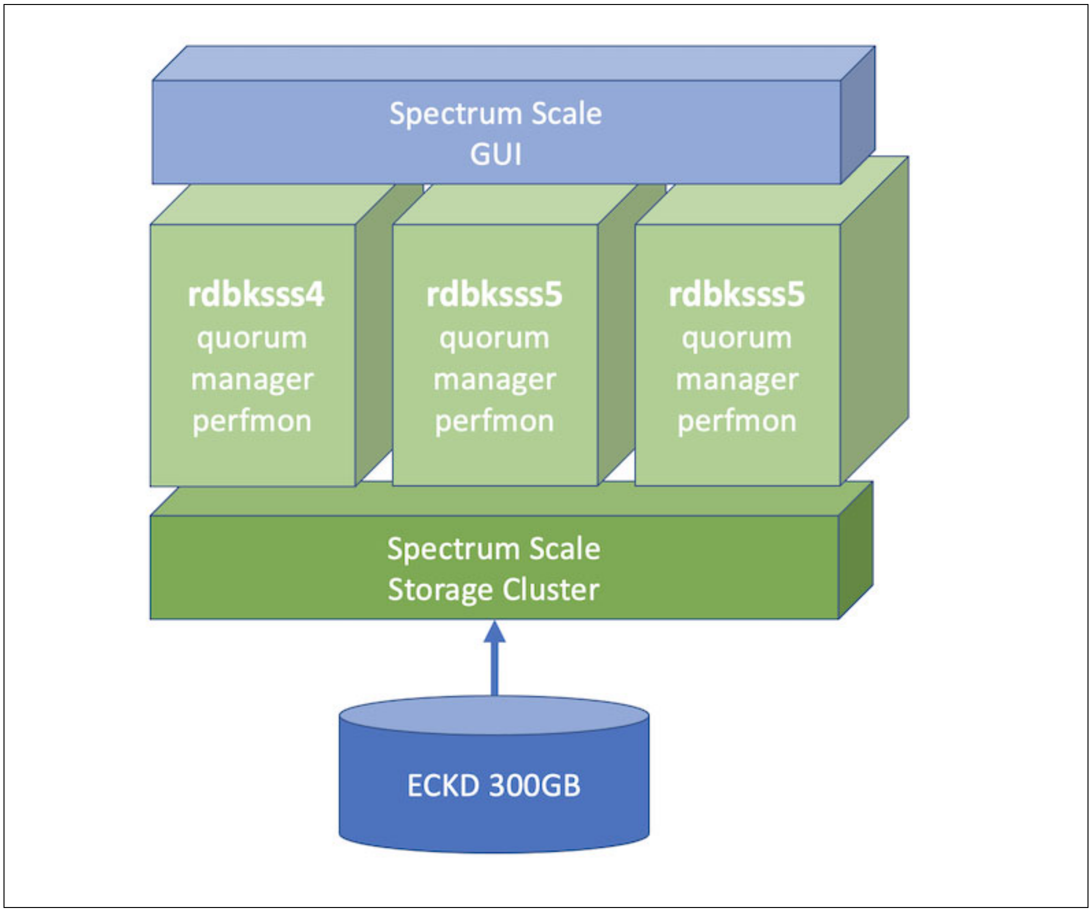



*Figure 4-4 Spectrum Scale Storage Cluster*

The installation of this cluster was completed by using the installation toolkit and the procedure that is described at this IBM Documentation web page.

**Note:** During the installation of IBM Spectrum Scale, we encountered some issues with the SSH key prerequisites for the installation. It is recommended to create the keys with the -PEM option: `ssh-keygen -p -N "" -m pem -f /root/.ssh/id_rsa`.

After the cluster is deployed, the GUI can be accessed and used to generate the needed file systems.

Figure 4-5 on page 49 shows the IBM Spectrum Scale GUI.

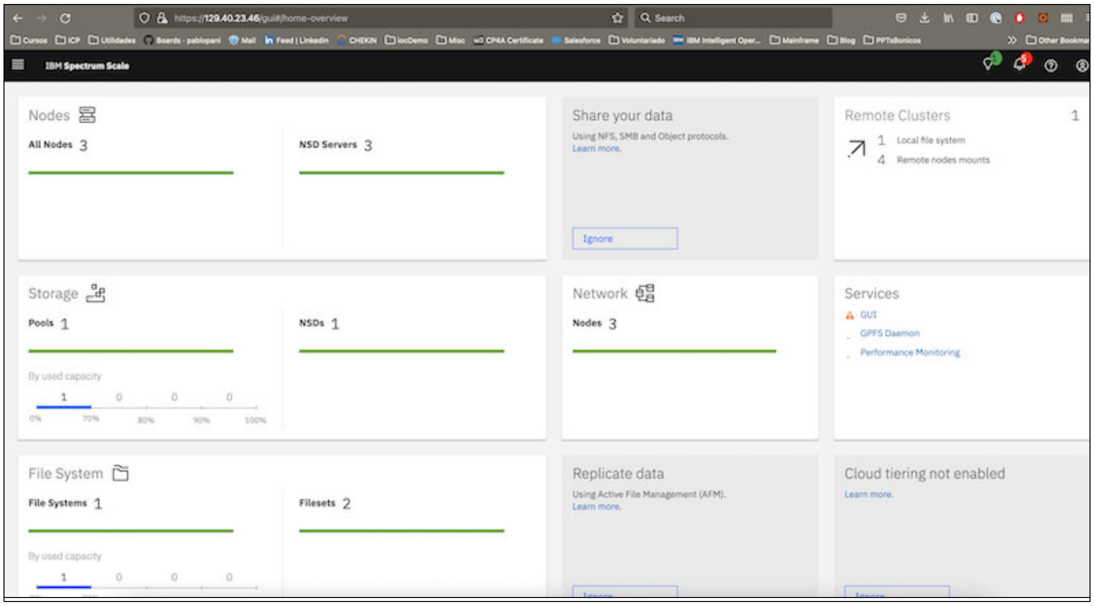



Figure 4-5 Spectrum Scale GUI

A file system `fsrdbkss` of approximately 300 GB was created to provide storage to IBM Client Innovation Center and Red Hat OpenShift.

## IBM Spectrum Scale CNSA in our environment

IBM Spectrum Scale Container Native Storage Access provides storage to the containers on Red Hat OpenShift. It was installed by using the guide that is available at this IBM Documentation web page.

The CNSA Operator then deploys pods in each node that is charged with mounting the file systems that are defined in our IBM Spectrum Scale cluster.

> **Note:** A minimum of three worker nodes are required for CNSA installation.

After completing the installation, the pods that are shown on Figure 4-6 are deployed.

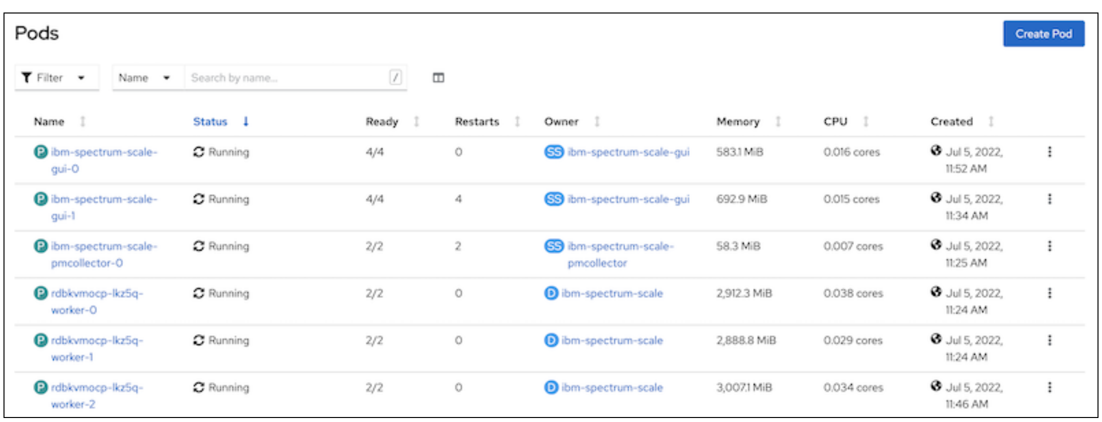



Figure 4-6 CNSA pods on Red Hat OpenShift

Apart from the pods in each node, a GUI and a performance monitor collector are also deployed. The GUI can be accessed through a route in Red Hat OpenShift, which makes it easier to manage the Spectrum Scale CNSA deployed.

After the cluster is correctly set up, a sample storage class is created.

### Red Hat OpenShift Data Foundation in our environment

Red Hat OpenShift Data Foundation or ODF also was installed along IBM Spectrum Scale CNSA to provide software-defined storage to the containers. I t was installed through the Operator Marketplace and configured by using a wizard that easily allows the user to define the storage and storage classes that are needed for Red Hat OpenShift.

Our lab environment was built with ODF internal storage services, which means that the resources that are needed for ODF were deployed internally on Red Hat OpenShift and not as an external Ceph cluster or IBM FlashSystem®.

To provision the internal services, some extra volumes must be attached to the Red Hat OpenShift worker nodes. This process was done by using IBM Cloud Infrastructure Center storage provisioning (see Figure 4-7).

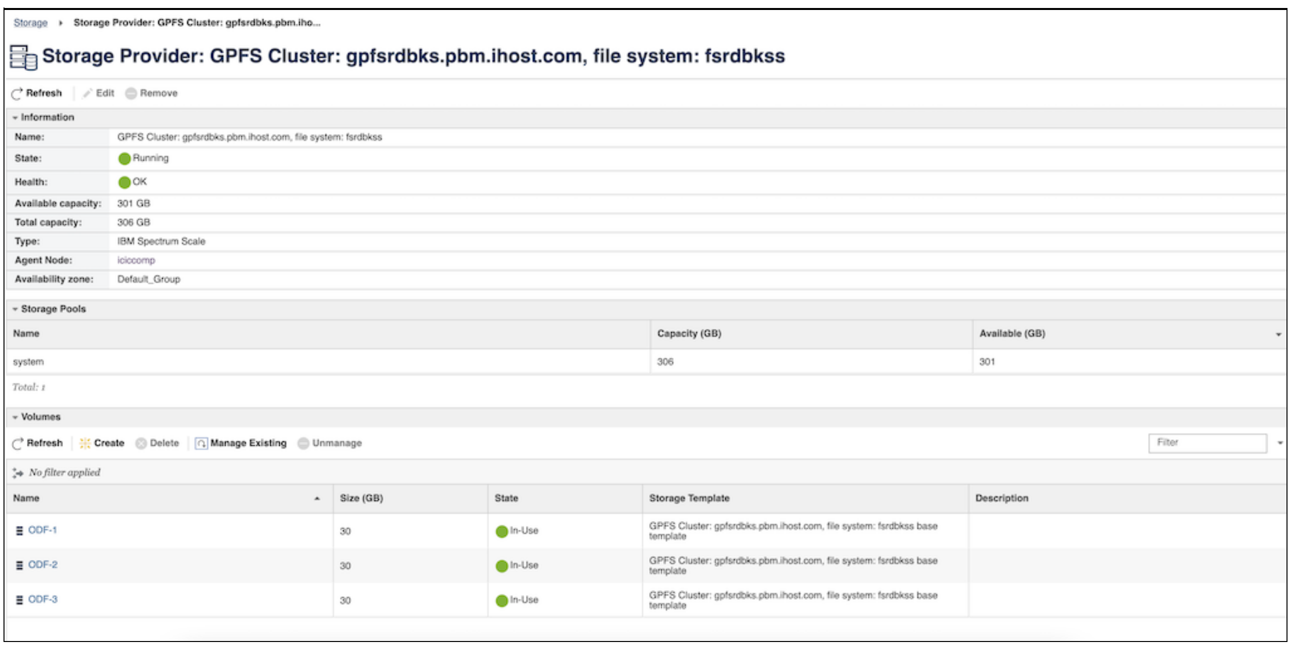



*Figure 4-7 ODF Storage Provisioning*

For this IBM Redbooks publication, three disks with 30 GB each were deployed, which is more than enough space to host the sample applications.

After these disks are correctly attached to each worker node, ODF installer recognizes the disks, and the internal cluster can be deployed (see Figure 4-8).

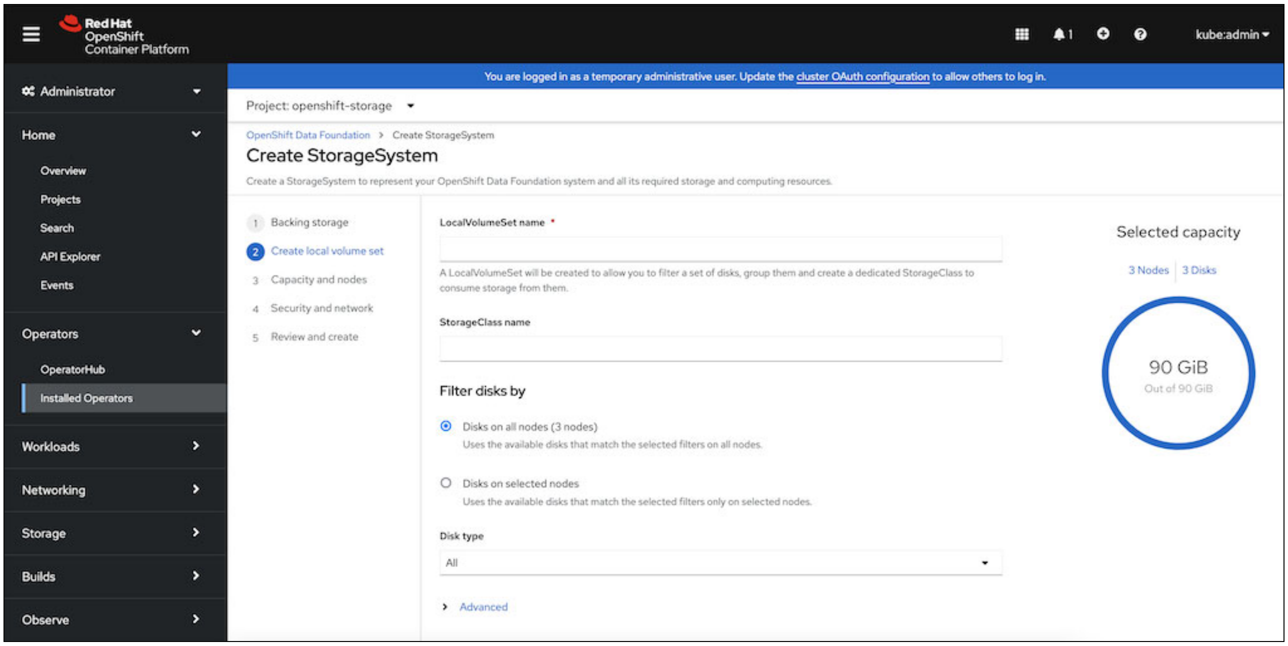



*Figure 4-8 Disk Recognition*

## 4.1.2 Storage for the private cloud

You can use external storage on IBM Cloud Infrastructure Center and the alternatives to this storage: ECKD and FCP.

## 4.2 Storage for containers

Containers by design are stateless, which means that if no persistent storage is configured, all data that is written after the container was deployed is deleted. This stateless concept is not valid for all kinds of containers because many of them require some sort of persistence to avoid data loss during the container lifetime or to set-up a specific configuration.

Red Hat OpenShift supports many types of storage including the following examples that allows the user to choose the kind of storage that best fits for each architecture.

- Ephemeral Storage

  This type of storage is the default storage that is chosen in Red Hat OpenShift if no persistence is required. It deploys the container without the use of any real persistence. All data that is generated in the container is deleted after the container is destroyed.

- Persistent Storage

  Data can be persisted on Red Hat OpenShift by using several methods. It can be a single file, a file system, a block device, an object storage, and so on. The most common methods are used to define storage in Red Hat OpenShift:

  – ConfigMaps: Configuration Maps are objects that are defined in Red Hat OpenShift that are commonly used for defining a configuration state in a container. Defining a configuration state in a container helps when multiple containers with different configurations are deployed because it avoids the need to build the container with a predefined configuration.

    A configuration map can be mounted as a file inside a container as a read-only file.

  – Secrets: The concept of secrets in Red Hat OpenShift is similar to ConfigMaps, with the exception that secrets are hold sensitive information, such as passwords or certificates. They also can be mounted as a file in a container as a read-only file.

  – PVs: These volumes and PVCs are the basic and most common objects in Red Hat OpenShift to provide storage for containers. PVs can support multiple types of storage (NFS, Cloud Storage, Local Storage, and so on), which is then claimed by a PVC and mounted in a container as Read/Write Once, Read Write Many, and Read Only.

Figure 4-9 shows how persistent storage is available to an application through PV and PVC.

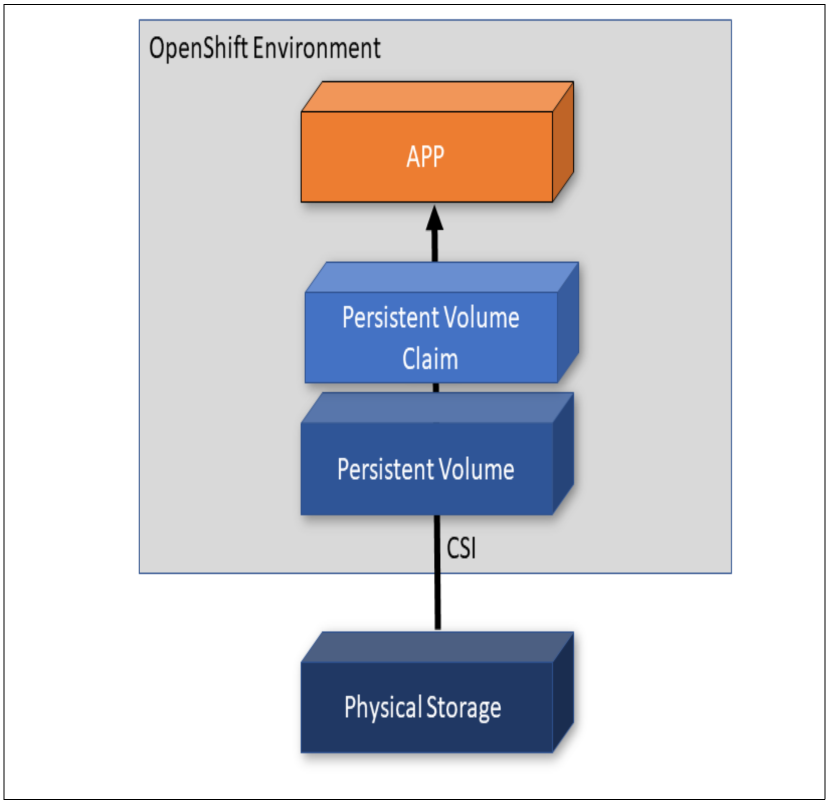



*Figure 4-9 Red Hat OpenShift persistence stack*

Because of Container Storage Interface (CSI), multiple vendors can integrate their storage provider solutions with Red Hat OpenShift to link PVs with volumes on a Storage solution. For the purposes of this IBM Redbooks publication, the Red Hat OpenShift Data Foundation and IBM Spectrum Scale are analyzed for Z Systems.

## 4.2.1 Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation is a software-defined storage solution for containers. It is a service build specifically for data and storage management on Red Hat OpenShift that eases the way the containers access to storage.

Red Hat OpenShift Data Foundation runs anywhere Red Hat OpenShift does, such as on-premises or public cloud, but also different architectures, such as x86, Z Systems, and Power.

It offers a simplified access for the user that is integrated with the Red Hat OpenShift console and prepared to provide data storage for all the Red Hat OpenShift applications. It also offers a consistent user experience, running as a service inside Red Hat OpenShift.

Administrators can consistently manage multiple environments across cloud footprints. Red Hat OpenShift Data Foundation is highly prepared for scalability and it can be dynamically provisioned across any new environment.

Red Hat OpenShift Data Foundation's most common use cases include the following examples:

- Data at rest, such as databases
- Data in motion. such as message brokers or serverless applications
- Data in action, such as machine learning and AI applications
- Any other stateful applications

Figure 4-10 shows different use cases for each of the workloads that can be deployed on Red Hat OpenShift.

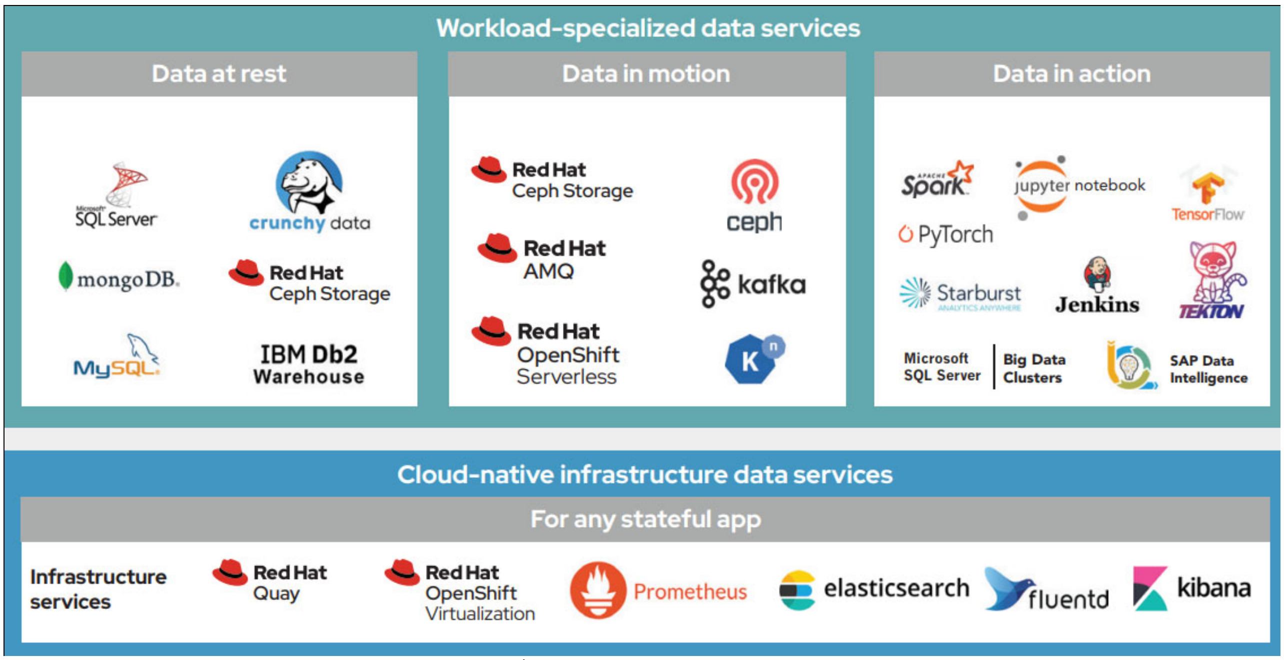



*Figure 4-10 Workload-specialized data services*[1]

The platform is based on open-source technology that is built upon technology, such as the following examples:

- Ceph: Ceph is an open-source, software-defined storage solution that implements object storage on a single distributed cluster with interfaces for object, block, and file storage.
- Nooba: Nooba is a data service for cloud environments that simplifies data administration by connecting to any of the storage silos from private and public clouds.
- Rook: Rook is an orchestration toll that helps to run Ceph inside a Red Hat OpenShift cluster. It turns distributed storage systems into self-managing, self-scaling, and self-healing storage services.

[1] https://www.redhat.com/en/resources/openshift-data-foundation-datasheet

Figure 4-11 shows the general architecture for Red Hat OpenShift Data Foundation.

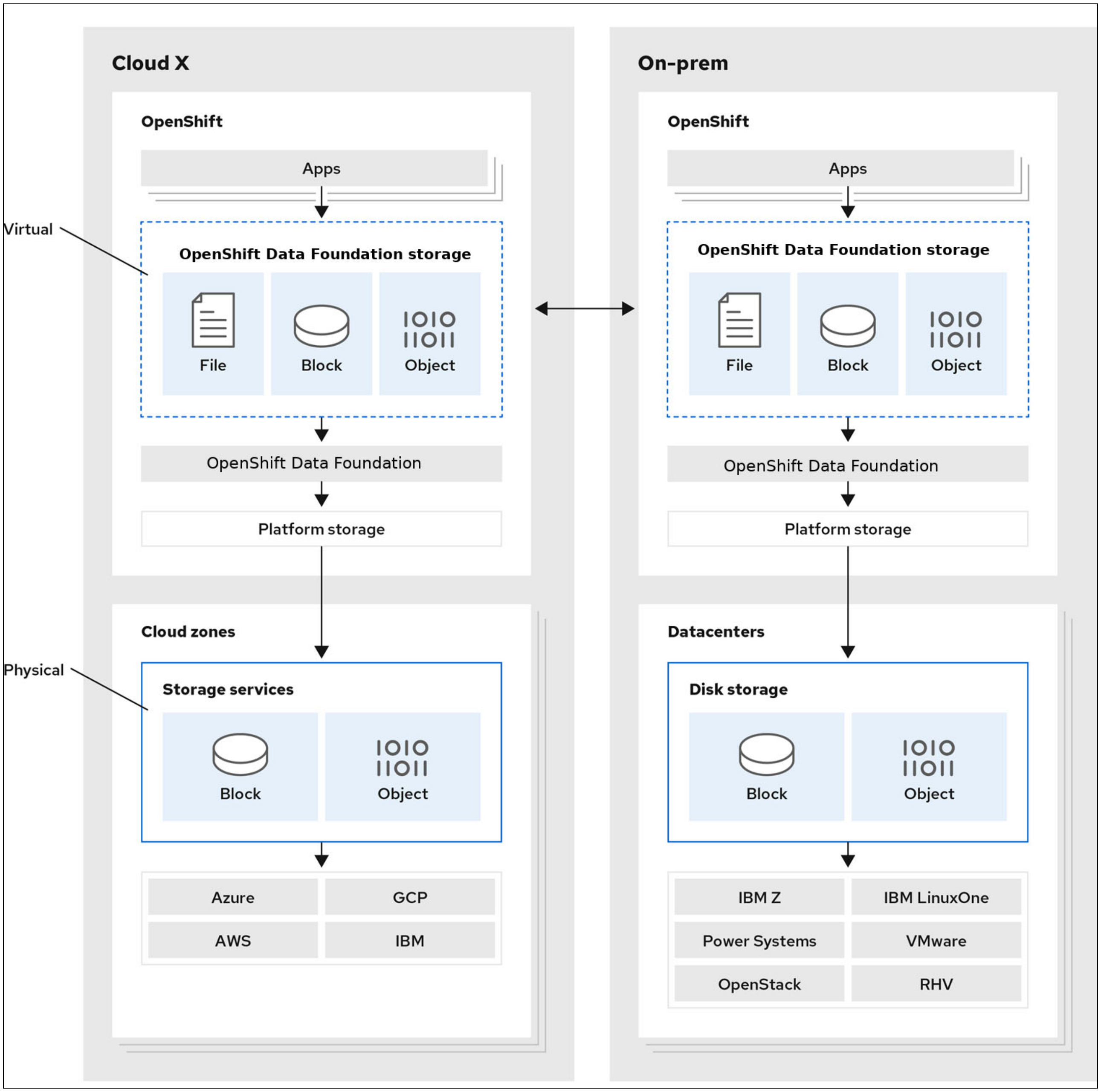



*Figure 4-11 Red Hat OpenShift Data Foundation architecture*

Red Hat OpenShift Data Foundation is deployed as three main operators, which include the administrative tasks and custom resources so that these tasks can be easily automated. The administrator must define only the wanted end state of the cluster.

The following operators ensure that the cluster is in that state or progressing to it:

- Red Hat OpenShift Data Foundation operator codifies and enforces the recommendations and requirements of an Red Hat OpenShift Data Foundation deployment.
- Rook-Ceph operator automates the packaging, deployment, management, upgrading, and scaling of persistent storage and services.

- MCG operator automates the services of the Multicloud Object Gateway service. It also creates an object storage class and services object bucket claims that are made against it.

Flexibility is core of Red Hat OpenShift Data Foundation, which is why the following methods are available with which it can be deployed:

- Internal deployment is entirely done within Red Hat OpenShift Container Platform. It includes all the benefits of an Operator-based deployment and management by using a local storage operator and local storage devices. It can be fully managed by using a GUI that is integrated with the Red Hat OpenShift Container Platform console.
- External deployment is used when Red Hat Ceph Storage services are running outside of the cluster. Red Hat OpenShift Data Foundation exposes these services as storage classes inside the cluster.

### 4.2.2 IBM Spectrum Scale container native storage access

IBM Spectrum Scale container native is a containerized version of IBM Spectrum Scale. IBM Spectrum Scale container native allows the deployment of the cluster file system in a Red Hat OpenShift cluster.

By using a remote mount attached file system, the container native deployment provides a persistent data store to be accessed by the applications through the IBM Spectrum Scale CSI driver by using PVs.

IBM Spectrum Scale container native with Red Hat OpenShift Container Platform supports the following features:

- IBM Spectrum Scale node labels to establish node affinity automated customer-only cluster creation
- Automated remote file system mount for IBM Spectrum Scale Storage cluster
- Integrated IBM Spectrum Scale Container Storage Interface (CSI) driver for application persistent storage Automated deployment of IBM Spectrum Scale Container Storage Interface (CSI) driver
- IBM Spectrum Scale container native customer cluster node expansion on Red Hat OpenShift Container Platform
- Cluster monitoring by using Red Hat OpenShift Container Platform Liveness and Readiness probe
- Call Home
- Performance data collection
- Storage cluster encryption
- Rolling upgrade
- Automated IBM Spectrum Scale performance monitoring bridge for Grafana
- File audit logging (FAL)
- Compression
- On the storage cluster:
  - Quotas
  - ACLs
  - ILM support
  - File clones
  - Snapshots

- Internet Protocol network connectivity among cluster nodes
- Direct storage attachment on s390x, x86, and power servers
- Automatic quorum selection is Kubernetes topology aware

### 4.2.3 IBM Storage Suite for Cloud Paks

IBM Storage Suite for Cloud Paks complements and supports the deployment and effective operation of IBM Cloud Pak® solutions.

IBM developed a series of middleware tools that is called IBM Cloud Pak solutions to enhance and extend the functions and capabilities of Red Hat OpenShift. The following solutions are included:

- Cloud Pak for Data
- Business Automation
- Watson AIOps
- Integration
- Network Application
- Security

These solutions give enterprises the fully modular and easy-to-use capabilities that are needed to bring the mission critical workloads into modern, cloud-based environments.

To integrate storage resource management into any cloud-based application modernization initiative, IBM offers a flexible software-defined storage solution set that is called IBM Storage Suite for IBM Cloud Paks. This solution is based on members of the IBM Spectrum Storage family and offerings from Red Hat that brings enterprise data services to container environments with a comprehensive set of software-defined storage offerings to satisfy almost any workload requirement.

The IBM Cloud Paks combined with IBM Storage Suite for IBM Cloud Paks delivers a modular, easy-to-use, enterprise-ready data services foundation for containerized environments. This combination provides an open, faster, and more secure way to move core business applications to any cloud.

IBM Storage Suite for IBM Cloud Paks includes data resources for file, object, and block data, and services for data management. The suite offers the flexibility to choose between open-source Red Hat storage solutions or award-winning IBM Spectrum Storage solutions for file, block, and object storage.

Designed to simplify IBM Cloud Pak setup with an immediately available storage layer, the suite is Cloud Pak recommended, tested with IBM Cloud Pak and Red Hat OpenShift with ongoing security, compliance, and version compatibility.

IBM Storage Suite for IBM Cloud Paks provides a flexible menu of data management and data services that more transparently and comprehensively automate and enhance the integration of underlying storage and data management resources.

An overview of various components in IBM Storage Suite for IBM Cloud Paks is shown in Figure 4-12.

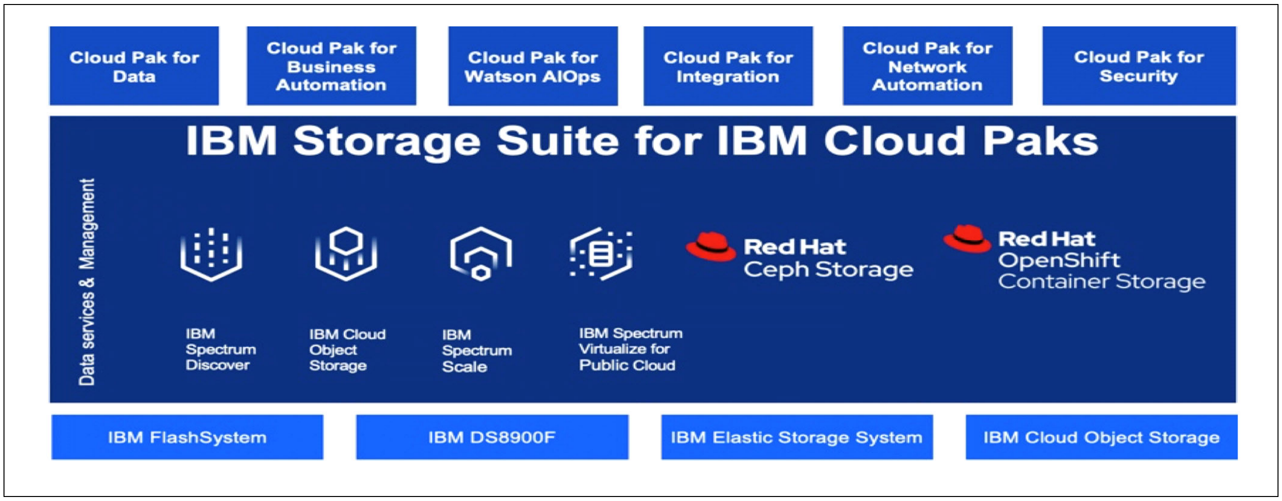



*Figure 4-12 IBM Storage Suite for IBM Cloud Paks*

Next, we describe the salient features of the components and when it is better to use each component.

## IBM Spectrum Scale

IBM Spectrum Scale provides high performance, simple scalability, and data access for edge to core to public cloud and the hybrid cloud optimized data center. It offers a full-featured set of data management tools, including advanced storage data optimization, global collaboration for data-anywhere access that spans storage systems, and geographic locations, with intelligent storage tiering.

With the new containerized IBM Spectrum Scale, Kubernetes containers have a simple way to provide faster access to data for containerized workloads.

IBM Spectrum Scale supports a range of application workloads at scale by using various access protocols and is effective in large, demanding environments.

Providing a high-performance interface to multivendor NFS or object data make IBM Spectrum Scale the choice when creating an AI information architecture for global enterprise access, including containerized workloads.

## IBM Spectrum Virtualize for Public Cloud

IBM Spectrum Virtualize for Public Cloud is the public-cloud-based counterpart of the software at the heart of IBM FlashSystem, IBM Spectrum Virtualize.

A new way is needed to modernize data center architectures and provide a single data fabric on a cloud native architecture, such as Red Hat OpenShift that can extend the infrastructure into new and modern AI and public cloud environments.

The data fabric can provide a single control point with consistent enterprise-class performance and management and a consistent user experience between on-premises clouds or public clouds, such as Amazon Web Services (AWS) and IBM Cloud.

IBM Spectrum Virtualize for Public Cloud and IBM FlashSystem are container-ready storage for Red Hat OpenShift that can free enterprises from silos and modernize their current infrastructure through virtualization

### IBM Cloud Object Storage

IBM Cloud Object Storage is a highly scalable cloud storage solution for unstructured data that provides on-premises and cloud-based dedicated services. It enables enterprises to store and manage massive amounts of data efficiently and securely, with over "fifteen-nines" of system durability and "eight nines" of availability with always on data and no forklift upgrades required.

IBM Cloud Object Storage uses an innovative approach for cost-effectively storing large volumes of unstructured data. It delivers the capabilities that are required to provide continuous access to data assets to improve research outcomes, decision making, and responsiveness to business, regulatory, or legal demands.

### IBM Spectrum Discover

IBM Spectrum Discover is a multi-source data catalog that automatically and continuously indexes objects and files whenever changes are made by using the metadata in real time. The result is a powerful and customizable database with a user-friendly interface that allows users to locate and identify the most relevant data regardless of its type or location.

By using a simple SQL query command or actionable API scripts or commands, users are empowered with comprehensive insight into the data in a fast and efficient manner.

IBM Spectrum Discover also can be used to create custom tags and policy-based workflows to orchestrate content inspection and activate data in AI, machine learning (ML), and analytics workflows.

IBM Spectrum Discover can be used for faster AI analysis, compliance classification, image and video indexing, identifying personal data, AI data pipeline integration, real-time data discovery, and provide new insights to optimize data and find bad or duplicate data.

Data sources include IBM Spectrum Scale, IBM COS, AWS S3, NFS. or SMB data sources, including Netapp and Isilon and Red Hat (Ceph and Red Hat OpenShift Container Storage).

### Red Hat OpenShift Container Storage

Red Hat OpenShift Container Storage is integrated with Red Hat OpenShift Container Platform to deliver persistent container storage services for all types of data, including block, file, and object. Red Hat OpenShift Container Storage presents a consistent interface for administrators, users, and developers, no matter the underlying infrastructure (public cloud, private cloud, or on- premises virtualized or bare metal). It is deployed by using operators for dramatically simplified day 0 installation and day 1 configuration and streamlined day 2 operational management.

### Red Hat Ceph Storage

Red Hat Ceph Storage is an open, massively scalable storage solution for modern workloads, such as AI/ML and data analytics, media and content repositories, cloud infrastructure, and backup and restore systems. It delivers software-defined storage on your choice of industry-standard hardware.

With block, object, and file storage combined into one platform, Red Hat Ceph Storage efficiently and automatically manages all your data and scales to support 100s of petabytes.

**Note:** Based on the storage requirements in your environment, IBM Storage Suite for IBM Cloud Paks offers the following options for capacity:

- Standard Edition delivers up to 2 TB of storage capacity per VPC licensed.
- Capacity Edition delivers up to 20 TB of storage capacity per VPC licensed.

IBM Spectrum Discover and Red Hat OpenShift Container Storage are available for each deployment and do not count toward capacity entitlement. Buy what you need when you need it and mix and match your data service capacities as business dictates; that is, block, file, or object.

### 4.2.4 Comparing IBM Storage Suite for IBM Cloud Paks and IBM Spectrum Storage Suite

IBM Storage Suite for IBM Cloud Paks and IBM Spectrum Storage Suite include the following main similarities and differences:

- Both contain Spectrum Scale that work well for Red Hat OpenShift Platform on Z Systems.
- Both contain IBM Cloud Object Storage for S3 data storage and access for eligible hybrid cloud workloads.
- IBM Storage Suite for IBM Cloud Paks contains Red Hat OpenShift Data Foundation/OCS Advanced Edition (if purchased before 2022).
- Spectrum Storage Suite includes Spectrum Protect Plus for container data protection (generally available on Z Systems on 29 June 2022).

5

# Aspects of cloud applications

In this chapter, we discuss the various aspects of cloud applications.

This chapter includes the following topics:



## 5.1 Application architecture

In our lab, we deployed an open-source, lightweight, and microservices-based application that is called `Voting app`. It is cross platform and can be deployed on any architecture.

We use this application as an example to demonstrate the benefits of deploying this containerized application in a hybrid cloud environment, which is powered by Red Hat OpenShift and IBM Cloud Infrastructure Center on-premises and Z Systems.

For more information about the public repository with the application source code, see this GitHub web page.

The application provides the user with a choice to vote for any of two given options (such as cats versus dogs).

The application is a collection of five microservices that cooperate to become the voting application. The microservices architecture is shown in Figure 5-1.

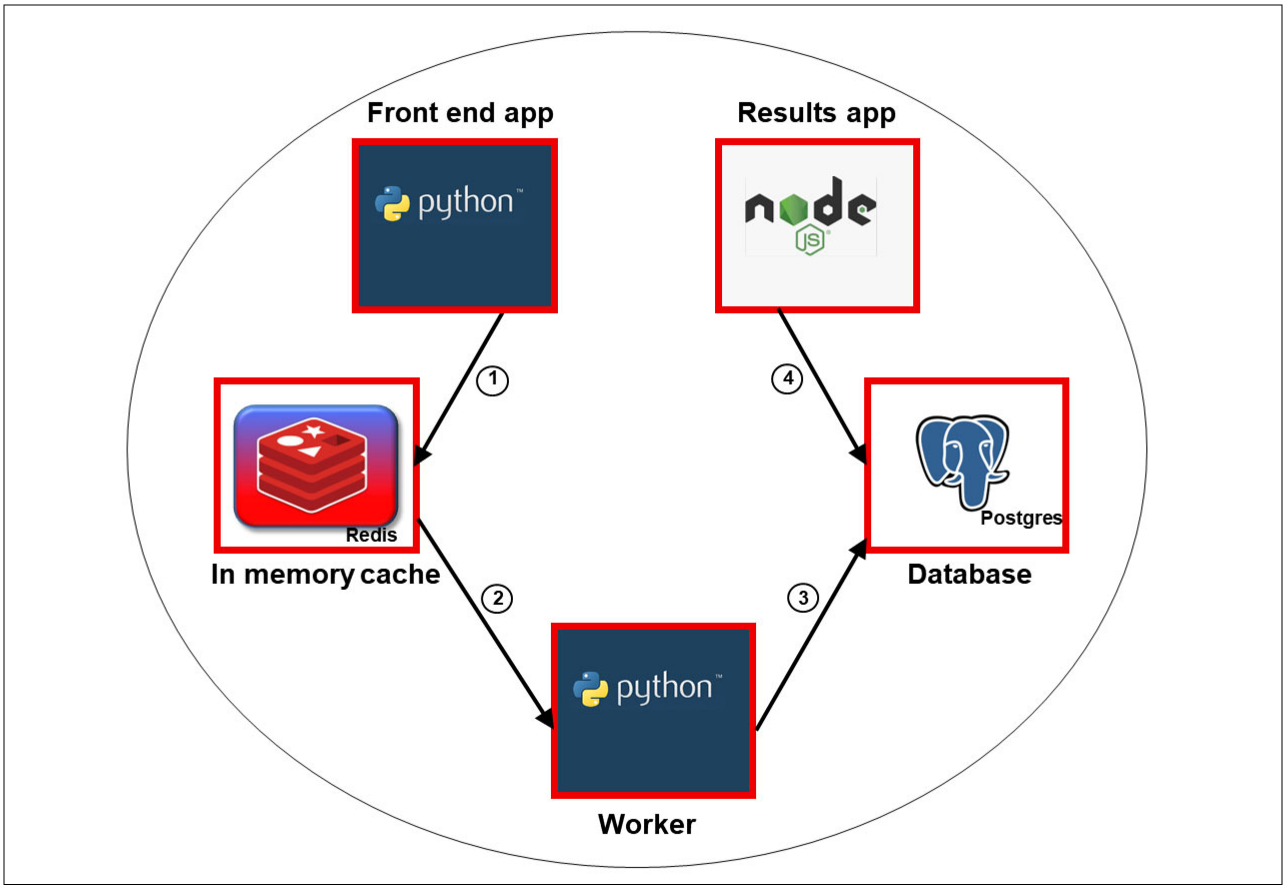



*Figure 5-1 Application architecture*

The web front end consists of a Python microservice that displays the options from which users choose. When the user interacts with the application, the information (votes) is sent to the Redis microservice (1 in Figure 5-1).

Redis serves as an in-memory cache holding the votes received by the Python web microservice. Running in a background is another microservice (also written in Python) that takes the votes from Redis (2 in Figure 5-1 on page 62) and stores them in a Postgres database (3 in Figure 5-1 on page 62).

Finally, a `Node.js` microservice shows the voting results as they are accumulated (4 in Figure 5-1 on page 62 in the Postgres database. The application follows the notation of microservices-based architecture and its components can be scaled individually. In fact, each component can be switched with an alternative one, without affecting the overall application.

## 5.2 Deployment

We deployed the selected application on an Z Systems hybrid cloud that is powered by the Red Hat OpenShift platform and the IBM Cloud Infrastructure Center. The Red Hat OpenShift platform provides the tools and frameworks for developers, with which they can create the pipelines and develop and deploy applications directly to the cloud and many other DevOps aspects.

To deploy the application, you need the GitHub personal access token that is used for generating a secret. For more information, see this GitHub web page.

Complete the following steps to start the deployment:

1. Set the following environmental variables:
   - `$PROJECT`
   - `$GIT_REPO`
   - `$GIT_TOKEN`
2. Create the project by using the following command:
   ```
   oc new-project ${PROJECT}
   ```
3. Create the secret by using the following command:
   ```
   oc create secret generic git-token --from-file=password=${GIT_TOKEN}
   --type=kubernetes.io/basic-auth
   ```
4. Import images by using the following commands:
   ```
   oc import-image rhel8/nodejs-12 --from=registry.redhat.io/rhel8/nodejs-12
   --confirm
   ```
   ```
   oc import-image ubi8/python-38 --from=registry.redhat.io/ubi8/python-38
   --confirm
   ```
5. Deploy the Postgres database and wait for the service to be ready by using the following commands:
   ```
   oc new-app --name new-postgresql --template=postgresql-persistent \
    --param=DATABASE_SERVICE_NAME=new-postgresql \
    --param=POSTGRESQL_USER=admin \
    --param=POSTGRESQL_DATABASE=db \
    --param=POSTGRESQL_PASSWORD=admin \
    --param=POSTGRESQL_VERSION=latest
   ```

6. Deploy the Redis service by using the following commands:

```
oc new-app --name new-redis --template=redis-persistent \
 --param=DATABASE_SERVICE_NAME=new-redis \
 --param=REDIS_PASSWORD=admin \
 --param=REDIS_VERSION=latest
```

7. Deploy the Python worker by using the following commands:

```
oc new-app python-38:latest~${GIT_REPO} \
 --source-secret=git-token \
 --context-dir=worker-python \
 --name=voting-app-worker-py \
 -e DB_NAME="db" \
 -e DB_USER="admin" \
 -e DB_PASS="admin" \
 -e REDIS_PASSWORD="admin"
```

8. Deploy the voting application by using the following commands:

```
oc new-app python-38~${GIT_REPO} \
 --source-secret=git-token \
 --context-dir=/vote \
 --name=voting-app-py \
 -e REDIS_PASSWORD="admin"
```

9. Deploy the `Node.js` front-end application by using the following commands:

```
oc new-app nodejs-12:latest~${GIT_REPO} \
 --source-secret=git-token \
 --context-dir=result \
 --name=voting-app-nodejs \
 -e POSTGRES_CONNECT_STRING="postgres://admin:admin@new-postgresql/db"
```

10. Create the application routes by using the following commands:

```
oc create route edge demo-nodejs --service=voting-app-nodejs --port=8080
oc create route edge demo-py --service=voting-app-py --port=8080
```

Step 10 exposes the routes of the application's front ends so they can be accessed by using a web browser. In our environment, the following URLs were used:

- `https://demo-nodejs-rdbk-vote.apps.rdbkvmocp.pbm.ihost.com`
- `https://demo-py-rdbk-vote.apps.rdbkvmocp.pbm.ihost.com`

## 5.3 Observed results and conclusions

In our lab environment, we implemented the hybrid cloud on-premises environment for the application, which is powered by Red Hat OpenShift and the IBM Cloud Infrastructure Center. This platform enables greater flexibility for developing, deploying, and managing (Day 1 and Day 2 operations) containerized applications.

Hybrid cloud on-premises uses core Z Systems technologies with which application can scale in three dimensions (horizontally, vertically, or both). By doing so, it achieves superior performance and Service Level Agreements to a similar deployment in a public cloud, all within the highly secure Z Systems or IBM LinuxONE server.

A hybrid cloud on-premises on Z Systems provides all the means to integrate with core System of Records (IBM Db2, CICS®, IMS, and so on) by using the data gravity and reducing network latencies.

Powerful Z Systems hardware (IBM PR/SM) and software hypervisors (z/VM, KVM) work together to provide a highly virtualized platform for a mixed variety of workloads, which run securely isolated.

Such deployment adheres to the standards of responsible computing and minimizes the environmental impact, while still delivering the required services to users.

# Related publications

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

## Online resources

The following websites also are relevant as further information sources:

- Ansible-Automated OpenShift Provisioning on KVM on Z Systems/LinuxONE:

  https://ibm.github.io/Ansible-OpenShift-Provisioning/
- Installing Red Hat OpenShift on the IBM Cloud Infrastructure Center via user-provisioned infrastructure:

  https://github.com/IBM/z_ansible_collections_samples/tree/master/z_infra_provisioning/cloud_infra_center/ocp_upi
- Infrastructure-as-a-Service:

  https://www.ibm.com/cloud/learn/iaas
- IBM Cloud Infrastructure Center documentation:

  https://www.ibm.com/docs/en/cic
- Hardware and software requirements for z/VM system:

  https://www.ibm.com/docs/en/cic/1.1.5?topic=requirements-hardware-software-zvm-system
- Red Hat OpenShift Container Platform on Z Systems and IBM LinuxONE Reference Architecture:

  https://www.ibm.com/docs/linuxonibm/liaaf/lnz_r_rhocp.html
- Installing a cluster with z/VM on Z Systems and LinuxONE:

  https://docs.openshift.com/container-platform/4.10/installing/installing_ibm_z/installing-ibm-z.html
- Installing a cluster with RHEL KVM on Z Systems and LinuxONE:

  https://docs.openshift.com/container-platform/4.10/installing/installing_ibm_z/installing-ibm-z-kvm.html

## Help from IBM

IBM Support and downloads

**ibm.com**/support

IBM Global Services

**ibm.com**/services

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

SG24-8530-00

ISBN 0738460885

Printed in U.S.A.