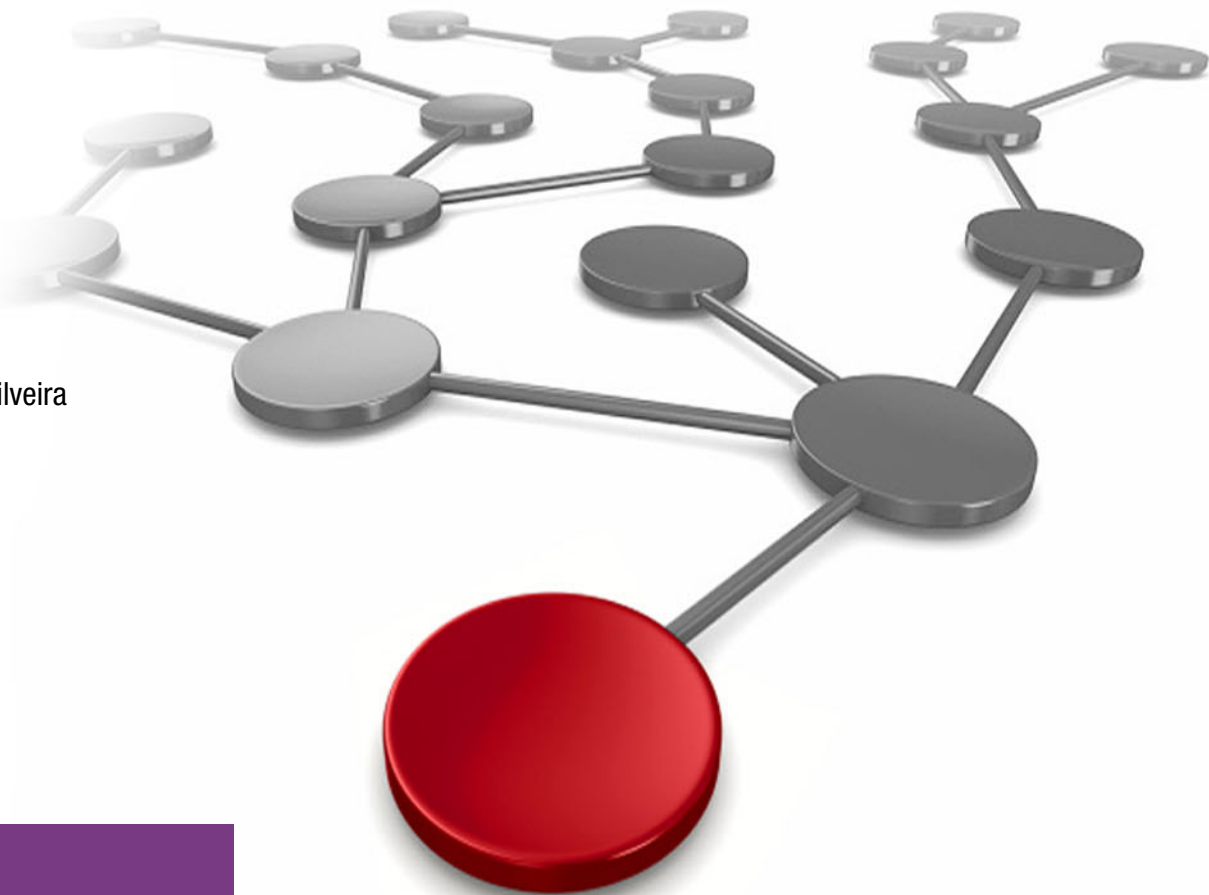


Practical Migration from x86 to Linux on IBM Z

Vic Cross
Youssef Largou
Eric Marins
Mauro Cesar de Souza
Phillip Wilson
Gary Wing
Anderson Augusto da Silveira



IBM Z



International Technical Support Organization

Practical Migration from x86 to Linux on IBM Z

September 2024

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

Second Edition (September 2024)

This edition applies to IBM Z z16, z/VM Version 7.2 and 7.3, Db2 Version 11.5, SUSE Linux Enterprise Server Version 15, Canonical Ubuntu Server 20.10 and 22.04, and Red Hat Enterprise Linux Version 8 and 9. Versions of other software components are incident to the versions available from the respective distributions referenced above.

© Copyright International Business Machines Corporation 2024. All rights reserved.

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

Contents

Notices	ix
Trademarks	x
 Preface	 xi
Authors	xi
Now you can become a published author, too!	xiii
Comments welcome	xiii
Stay connected to IBM Redbooks	xiii
 Chapter 1. Benefits of workload migration to Linux on IBM Z	 1
1.1 Benefits	2
1.2 Reasons to select Linux on IBM Z	2
1.2.1 Best of Enterprise Linux and Open Source	3
1.2.2 IBM Z strengths	4
1.3 Enabling a microservices architecture on Linux on IBM Z	6
1.4 Financial benefits of a migration	7
1.5 Cloud computing blueprint for Linux on IBM Z	15
1.5.1 Cloud Solutions on Linux on IBM Z	15
1.6 Linux on IBM Z flexibility and sustainability	18
 Chapter 2. Analyze and understand	 21
2.1 Total cost of ownership analysis	22
2.2 Choosing workloads to migrate	23
2.2.1 Ideal use cases for Linux on IBM Z	23
2.2.2 Ideal use cases for Red Hat OpenShift on IBM Z	24
2.3 Financial benefits of a migration	24
2.4 Overview of migration steps	26
2.5 Analysis of how to size workloads for migration	27
 Chapter 3. Virtualization concepts	 31
3.1 The demand for virtualization	32
3.2 Typical x86 virtualization	32
3.3 IBM Z virtualization options	33
3.3.1 IBM z/VM	34
3.3.2 KVM on IBM Z	37
3.3.3 PR/SM and DPM	38
3.4 Single system image and live guest relocation	39
3.5 z/VM hypervisor components	41
3.5.1 Control program	41
3.5.2 Conversational Monitor System	41
3.5.3 Log-On Wave for IBM Z	42
3.5.4 IBM Cloud Infrastructure Center	42
3.6 Virtualized resources	43
3.6.1 Virtualized CPU	43
3.6.2 Virtualized disk	44
3.6.3 Virtualized memory	45
3.6.4 Virtualized network	47
 Chapter 4. Migration process	 49

4.1 Stakeholder definitions	50
4.1.1 Business stakeholders	50
4.1.2 Operational stakeholders	50
4.1.3 Security stakeholders	53
4.2 Identify the stakeholders	53
4.3 Assembling the stakeholders	54
4.4 Migration methodology	55
4.4.1 Pre-assessment	55
4.4.2 Define success criteria	56
4.4.3 Finalizing the new environment.	56
4.4.4 Pilot proof of concept	57
4.4.5 Decision to migrate	57
4.4.6 Resource estimation	57
4.4.7 Actual migration	58
4.4.8 Verification testing.	58
4.4.9 Checking against success criteria.	58
Chapter 5. Migration planning	61
5.1 Migration project time commitments	62
5.2 Project definition	63
5.3 Planning worksheets	63
5.3.1 Software products and tooling worksheet.	63
5.3.2 Application implementation worksheet	63
5.3.3 Application flows worksheet	65
5.3.4 Training worksheet	65
5.3.5 Hardware planning worksheet.	65
5.3.6 Firewall planning checklist	66
5.3.7 Security and privacy worksheet	67
5.3.8 Sustainability worksheet	69
Chapter 6. Migration analysis	71
6.1 Network analysis	72
6.1.1 Network facilities available on IBM Z and z/VM	72
6.1.2 Network migration overview	74
6.1.3 Helpful steps for a network migration	84
6.2 Storage analysis	84
6.2.1 Data migration.	84
6.2.2 Linux on IBM Z: pre-installation considerations	88
6.3 VMware to KVM migration options	94
6.4 Application analysis.	95
6.4.1 Application architecture overview	95
6.4.2 Why migrate applications	96
6.4.3 Which applications can be migrated	97
6.4.4 Selecting an application for migration to Linux on IBM Z	97
6.4.5 Best-suited application for migration.	98
6.4.6 Other software	99
6.4.7 Selecting an application for a proof of concept.	100
6.4.8 Applications not supported on Linux on IBM Z	101
6.4.9 Application interdependencies	101
6.4.10 Successful application migration.	101
6.4.11 Special considerations for migrating a Java application	101
6.4.12 Special considerations for migrating C++ applications	103
6.4.13 Middleware, libraries, and databases	104

6.4.14	Helpful steps for an application migration	104
6.5	Database analysis	105
6.5.1	Before database migration	105
6.5.2	Migrating a single instance	105
6.5.3	Migrating multiple instances	105
6.5.4	Technical considerations	107
6.5.5	Migrating DB2 and Oracle from x86 to IBM Z	110
6.5.6	Tips for successful migration.	111
6.6	Backup analysis	112
6.6.1	Introduction to backup and archival concepts	112
6.6.2	KVM backup	113
6.6.3	z/VM backup	113
6.6.4	Linux backup	114
6.6.5	Migrating backed-up and archived data	114
6.6.6	General archival migration considerations	115
6.7	Security analysis	116
6.7.1	Security migration overview	116
6.7.2	Understanding the z/VM foundation	117
6.7.3	Hardening the base Linux on IBM Z	119
6.7.4	Code and application analysis	120
6.7.5	Security issues	120
6.7.6	Dependencies	120
6.7.7	Checking user input	121
6.7.8	Planning for updates when migrating code	121
6.7.9	Networking	121
6.7.10	Logging and recording events	121
6.7.11	Escalations of authority	122
6.7.12	Security test plan and peer review	122
6.7.13	Availability and accountability	122
6.7.14	Accountability analysis	123
6.7.15	Data integrity and confidentiality	124
6.7.16	Confidentiality analysis	125
6.7.17	Security change management	126
6.7.18	Enterprise authentication options	126
6.7.19	Integrated Cryptographic Service Facility	127
6.8	Operational analysis	127
6.8.1	The operational environment	128
6.8.2	Operational migration tasks	128
6.8.3	Single system image and live guest relocation	129
6.8.4	z/VM and virtual machine management products	129
6.9	Disaster recovery and availability analysis	130
6.9.1	Availability analysis	131
6.9.2	Single points of failure	131
6.9.3	IBM Z features for high availability	132
6.9.4	Availability scenarios	132
6.9.5	Linux-HA Project	140
6.9.6	HA add-ons provided by SUSE and Red Hat	140
6.9.7	Understanding the availability requirements of your applications	141
6.9.8	Service level agreements	141
6.9.9	The cost of availability	142
6.10	Linux on IBM Z cloud management	142
Chapter 7.	Deployment of workloads	145

7.1	Deciding between containers and VMs	146
7.2	Deploying HA clustering	148
7.3	Setting up Docker	148
7.3.1	Containers on Red Hat	148
7.3.2	Installing and configuring Docker	149
7.3.3	Testing Docker	155
7.4	Deploying MongoDB on Linux on IBM Z	156
7.4.1	Deploying MongoDB as a Docker container	156
7.4.2	Deploying MongoDB by using package manager	162
7.5	Deploying MediaWiki and MySQL	165
7.5.1	Analysis and planning	165
7.5.2	Installing the LAMP stack on Red Hat Enterprise Linux	166
7.5.3	Starting and testing LAMP components	166
7.5.4	Installing MediaWiki	170
7.5.5	Migrating iSCSI disks containing MySQL and MediaWiki	170
7.6	Deploying OpenLDAP	175
7.6.1	Analysis and planning	175
7.6.2	Installing LDAP software	176
7.6.3	Configuring the OpenLDAP service	176
7.6.4	Export OpenLDAP data from x86 server	180
7.6.5	Import OpenLDAP data to Linux on IBM Z	180
7.6.6	Verify that OpenLDAP is working	181
7.7	Deploying central log server	182
7.7.1	Analysis and planning	182
7.7.2	Initial configuration	183
7.7.3	Migrating with syslog-ng	189
7.7.4	Viewing logs: Grafana Loki	189
7.8	Deploying Samba	206
7.8.1	Installing Samba software	206
7.8.2	Configuring SAMBA	206
7.9	Deploying Terraform	209
7.10	Deploying Apache Kafka	211
7.11	Deploying Validated Open Source Software	216
7.12	Containerized Workloads	217
7.12.1	Solutions for container-based workloads on IBM Z	217
7.12.2	Running a container on Linux on IBM Z	218
7.12.3	Building a container for Linux on IBM Z	223
Chapter 8.	Hands-on migration	227
8.1	Setting up the system	228
8.1.1	Software products and tools worksheets	228
8.1.2	Hardware worksheet	228
8.2	Migrating Db2 and its data	229
8.2.1	Preliminary migration steps	230
8.2.2	Data migration using db2move and db2look	231
8.3	Migrating WebSphere Application Server Network Deployment cluster	233
8.4	Migrating WebSphere Application Server Liberty and Open Liberty	238
8.5	Migrating Fibre Channel devices	243
8.5.1	Zoning for FCP	244
8.5.2	FCP and multipath	244
8.5.3	FCP migration setup tasks	245
Chapter 9.	Postmigration considerations	247

9.1 Gaining acceptance	248
9.2 Performance measurement.	248
9.2.1 What is performance.	248
9.2.2 Choosing what to measure	249
9.3 Performance tuning.	250
Appendix A. Additional use case scenarios.	253
Telecommunication Company using Linux on IBM Z Crypto Capabilities for Cybersecurity	254
Healthcare industry: Mobile and Internet solution	255
Financial Data Provider industry Hybrid Cloud Modernization.	257
Appendix B. z/VM Express System Installation	259
About z/VM Express System Installation (ESI)	260
How does z/VM ESI help?	260
What does z/VM ESI provide?	260
Using z/VM ESI to install a hypervisor system on IBM Z	261
Complete the installation worksheet	261
Save the worksheet values (optional)	263
Load the LPAR from the ESI media	263
Provide installation parameters to the installer	264
Perform the installation	264
Shut down the installer and IPL the installed system	265
Using the ELAN to install ICIC	265
Related publications	267
IBM Redbooks	267
Online resources	268
Help from IBM	268

Notices

This information was developed for products and services offered in the U.S.A.

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, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: 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 states 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 supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

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 the names and addresses used by an actual business enterprise 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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®	IBM Cloud®	POWER®
BigInsights®	IBM Cloud Pak®	RACF®
CICS®	IBM FlashSystem®	Redbooks®
Cognos®	IBM Instana™	Redbooks (logo)  ®
DB2®	IBM Research®	System z®
Db2®	IBM Spectrum®	Think®
DS8000®	IBM Z®	Tivoli®
Envizi™	IBM z14®	WebSphere®
FICON®	IBM z16™	z/OS®
FlashCopy®	Instana®	z/VM®
GDPS®	OMEGAMON®	z16™
HyperSwap®	Open Liberty®	zEnterprise®
IBM®	Parallel Sysplex®	

The following terms are trademarks of other companies:

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

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

Open Mainframe Project, are trademarks of the Linux Foundation.

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

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

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

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

VMware, 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

There are many reasons why you would want to optimize your servers through virtualization using Linux on IBM® Z:

- ▶ Too many distributed physical servers with low utilization
- ▶ A lengthy provisioning process that delays the implementation of new applications
- ▶ Limitations in data center power and floor space
- ▶ High total cost of ownership (TCO)
- ▶ Difficulty allocating processing power for a dynamic environment

This IBM Redbooks® publication provides a technical planning guide and example for IT organizations to migrate from their x86 environment to Linux on IBM Z®. It begins by examining the benefits of migrating workloads to Linux on IBM Z. Here, we describe the workload-centric method of information technology and then discuss the benefits of migrating workloads to Linux on IBM Z.

Next, we describe total cost of ownership analyses and we guide you in understanding how to analyze your environment before beginning a migration project. We also assist you in determining the expected consolidation ratio for a given workload type.

We also describe virtualization concepts along with describing the benefits of migrating from the x86 environment to guests residing on an IBM z/VM® single system image with live guest relocation.

This IBM Redbooks publication walks you through a migration approach, including planning worksheets, and provides guidance for you in analyzing your own systems. We also discuss postmigration considerations such as acceptance testing of functionality and performance measurements.

Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Vic Cross is a Solutions Engineer in Brisbane, Australia. Vic works as part of the IBM Worldwide zAcceleration Team, helping customers in adopting new technologies, such as Red Hat OpenShift Container Platform on IBM Z and LinuxONE. Previously, Vic worked with IBM Systems Lab Services and IBM Systems, providing senior design and implementation expertise to IBM Z and IBM LinuxONE projects across Asia-Pacific. He holds a degree in Computer Science from Queensland University of Technology, and has spent most of his over-30-year IT career on the IBM Z and IBM LinuxONE platforms and their antecedents. Vic has written and contributed to several IBM Redbooks publications, including *SThe Virtualization Cookbook for IBM Z Volume 1: IBM z/VM 7.2*, SG24-8147, and *Securing Your Cloud: IBM z/VM Security for IBM z Systems and LinuxONE*, SG24-8353.

Youssef Largou is the founding director of PowerM, a platinum IBM Business Partner in Morocco. He has 22 years of experience in systems, HPC, middleware, and hybrid cloud, including IBM Power, IBM Storage, IBM Spectrum®, IBM WebSphere®, IBM Db2®, IBM Cognos®, IBM WebSphere Portal, IBM MQ, ESB, IBM Cloud® Pak, SAP HANA and Red Hat OpenShift. He has worked within numerous industries with many technologies.

Youssef is an IBM Champion 2020, 2021, 2022, 2023 and 2024, an IBM Redbooks Platinum Author and has designed many reference architectures. His company has been recognized as an IBM Beacon Award Finalist in Storage, Software-Defined Storage, and LinuxONE five times. He is a regular speaker at IBM Think®, IBM TechXchange and Common Europe Congress. He holds an engineering degree in computer science from the Ecole Nationale Supérieure des Mines de Rabat and an Executive MBA from EMLyon.

Eric Marins is a Principal IT Architect in Brazil, focused on hybrid cloud solutions, Infrastructure and Platform solutions and competencies, including High Availability, Disaster Recovery, Networking, Linux and Cloud. He has many years of experience working with, writing about IBM Z, Linux and Open Source topics. He has co-authored more than nine IBM Redbooks publications.

Mauro Cesar de Souza is a Senior Infrastructure Analyst in Brazil. He has 20 years of experience as a zVM administrator. He holds a bachelor's degree in IT from Unicamp. His areas of expertise include information security, high availability infrastructure, enterprise systems management. He has written extensively on information security and infrastructure management.

Anderson Augusto da Silveira Augusto is a Technical Specialist in Brazil and the Americas. He has 12 years of experience in virtualization, Linux on IBM Z and LinuxONE. Anderson holds a degree in Systems Development & Analysis from São Paulo University of Technology. His areas of expertise include Z Architecture, z/VM, KVM, Linux, Red Hat OpenShift Container Platform, and z Container Extensions. Anderson has written and contributed extensively on virtualization, microservices, and reference cases, as well as in IBM Redbooks publications including *IBM LinuxONE Resiliency*, SG24-8544.

Phillip Wilson is a Senior Technical Sales Specialist based at IBM in Coral Gables, FL, USA. He has 39 years of experience in a variety of fields which includes Microsoft Windows NT MCSE, Local Area Networks, VMware, x86 Servers, IBM Z, LinuxONE, and most recently Co-created an open source project to implement RedHat OpenShift on IBM Z KVM using LinuxONE.

Gary Wing is a seasoned mainframe specialist in Canada with over 25 years of experience in enterprise infrastructure. An alumnus of Queen's University and Cornell University, Gary has dedicated his career to mastering the complexities of mainframe architecture and operations. His areas of expertise include mainframe hardware, operating systems, and networking.

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

Lydia Parziale
IBM Redbooks, Poughkeepsie Center

Robert Haimowitz
IBM, Poughkeepsie Center

Stefan Schmitt,
IBM Germany

Christian Miemiec
IBM US

Mauro Souza, Rosana Rueda Elias, Beatriz Oliveira
Kyndryl, Brazil

Gayathri Gopalakrishnan
IBM India

Thanks to the authors of:

- ▶ *Practical Migration from x86 to LinuxONE*, SG24-8377-02, published in May, 2024:
Sandeep Batta, Youssef Largou, Pablo Paniagua, and Cecilia Vales
- ▶ *Practical Migration from x86 to Linux on IBM System z*, SG24-8217, published in September, 2014:
Lydia Parziale, Eduardo Franco, Craig Gardner, Berthold Gunreben, Tito Ogando, and Serkan Sahin
- ▶ *Practical Migration to Linux on System z*, SG24-7727, published in October 2009:
Lydia Parziale, Joseph Apuzzo, Saulo Augusto M Martins da Silva, Louis Henderson, Manoj Srinivasan Pattabhiraman, and Richard Sewell

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 ITSO 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, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Look for us on LinkedIn:

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

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



Benefits of workload migration to Linux on IBM Z

This chapter describes the benefits and reasons to migrate workloads to Linux on IBM Z. It also describes how Linux on IBM Z supports application modernization and the shift to a microservices architecture. It explains how any private, public, or hybrid cloud can benefit from Linux on IBM Z with a cloud computing blueprint.

This chapter includes the following topics:

- ▶ 1.1, “Benefits” on page 2
- ▶ 1.2, “Reasons to select Linux on IBM Z” on page 2
- ▶ 1.3, “Enabling a microservices architecture on Linux on IBM Z” on page 6
- ▶ 1.4, “Financial benefits of a migration” on page 7
- ▶ 1.5, “Cloud computing blueprint for Linux on IBM Z” on page 15
- ▶ 1.6, “Linux on IBM Z flexibility and sustainability” on page 18

1.1 Benefits

A significant benefit of migrating to Linux on IBM Z is that it allows organizations to break the link between the operating system (OS) and specific hardware platforms. This means that after your applications are running on Linux, you are no longer tied to a specific hardware platform. You have control over the choice of hardware platform that will support your application.

Linux is available on a large variety of computing platforms from set-top boxes and handheld devices to the largest mainframes. Linux running on IBM Z benefits from the hardware platform that includes a specialized processor, the Integrated Facility for Linux (IFL), cryptographic cards with dedicated RISC processors, and the bare metal hypervisor of IBM z/VM. Choosing to run applications on Linux gives you the freedom to select from the widest variety of hardware platforms. You can deploy applications on the best hardware to suit the needs of the application without having to run different operating systems.

A major benefit of Linux is that it is open source; the software is unencumbered by licensing fees and its source code is freely available. There are hundreds of Linux distributions available for almost every computing platform. Three enterprise distributions¹ of Linux are supported on Linux on IBM Z:

- ▶ [Red Hat: Red Hat Enterprise Linux \(RHEL\)](#)
- ▶ [SUSE Linux Enterprise Server](#)
- ▶ [Canonical Ubuntu Server](#)

Red Hat, SUSE and Ubuntu provide customers using Linux with various support options, including 24 x 7 support with one-hour response time worldwide for customers running production systems. As well as the Linux OS, SUSE, Ubuntu, and Red Hat offer a number of other open source products that they also fully support.

To simplify problem determination, IBM customers can contact IBM in the first instance and, if it is a new problem with Linux, IBM will work with Red Hat, Ubuntu, or SUSE to resolve the problem.

The increased interest and uptake of Linux resulted from its rich set of features, including virtualization, security, Microsoft Windows interoperability, development tools, a growing list of independent software vendor (ISV) applications, performance and, most importantly, its multiplatform support.

This multiplatform support allows customers to run a common operating system across all computing platforms, which will mean significantly lower support costs and, in the case of Linux, no incremental license charges. It also offers customers the flexibility of easily moving applications to the most appropriate platform. For example, many IT organizations choose Linux on IBM Z for the ability to scale databases across highly scalable hardware.

1.2 Reasons to select Linux on IBM Z

First announced in 1964, the IBM mainframe is the only computing system that has provided customers with a common architecture for more than 50 years. Today, as it has been for the last 50 years, the IBM Z is the most reliable and scalable computing platform available and it is the ideal platform for consolidating many hundreds of distributed servers.

¹ A Linux distribution is a complete OS and environment including compilers, file systems, and applications such as Apache (Web server), SAMBA (file and print), Sendmail (mail server), Tomcat (Java application server), MySQL (database), and many others.

In 2022, IBM introduced the multi frame model of the z16™ (z16 A01). In 2023 IBM added the Telum-based z16 single frame model (z16 A02) and the first rack mount IBM Z offering, the z16 AGZ. The rack mount offering enables you to install z16 technology in your own racks and co-locate with other technologies. The IBM z16™ portfolio of offerings brings the latest IBM z16 technology to businesses of all sizes and gives you the choice of what best fits the needs of your organization.

There are two current models of the IBM Z - A01, A02 (and AGZ). Both models share all of the characteristics that make the mainframe a uniquely powerful solution. The IBM z16 multi frame can scale to 200 configurable processors and 40 terabytes (TB) of memory while the IBM z16 single frame can scale to 68 configurable processors and 16 TB of memory.

Linux on IBM Z delivers the best of enterprise Linux on the industry's most reliable and highly scalable hardware. These systems are specialized scale-up enterprise servers that are designed exclusively to run Linux applications.

Linux on IBM Z provides the highest levels of availability (near 100% uptime with no single point of failure), performance, throughput, and security. End-to-end security is built in with isolation at each level in the stack, and provides the highest level of certified security in the industry.

IBM Secure Execution for Linux is a continuation and expansion of well-known security features of Linux on IBM Z. It supplements pervasive encryption, which protects data at-rest and data in-flight, to also protect data in-use. IBM Secure Execution for Linux is a hardware-based security technology that provides a trusted execution environment (TEE) for “Confidential Computing”. It provides scalable isolation for individual workloads to protect them from not only external attacks, but also insider threats. For more information, see [IBM Hyper Protect Platform: Applying Data Protection and Confidentiality in a Hybrid Cloud Environment](#).

The CryptoExpress cards (CEX) available on IBM z16 provide access to a FIPS 140-3 Level 4 Hardware Security Module (HSM), which is exploited by Hyper Protect Services to provide encryption key management services and quantum safe features for signing and encapsulation with Dilithium and Kyber.

Additionally, Linux on IBM Z facilitates transparent use of redundant processor execution steps and integrity checking, which is necessary in financial services industries. Linux on IBM Z servers typically enable hot-swapping of hardware, such as processors and memory. This swapping is typically transparent to the operating system, enabling routine repairs to be performed without shutting down the system.

1.2.1 Best of Enterprise Linux and Open Source

Linux on IBM Z provides the following benefits:

- ▶ Premium Linux experience with subsecond user response times and virtually unlimited scale.
- ▶ Broad portfolio of Open Source and other vendor products and tools delivered on the platform.
- ▶ Choice of Linux (RHEL, SUSE, and Ubuntu) and tools that best fit your environment.
- ▶ Eliminates risks by running Linux on the industry's most secure and resilient hardware platform.
- ▶ Easy integration of data and applications with existing IBM Z solutions.
- ▶ Increases operational IT efficiency.

1.2.2 IBM Z strengths

The strengths of the IBM Z are:

- ▶ Reliability
 - Redundant processors, I/O, and memory.
 - Error correction and detection.
 - Remote Support Facility.
- ▶ Availability
 - Fault tolerance.
 - Automated failure detection.
 - Non-disruptive hardware and software changes.
- ▶ Virtualization
 - High-performance logical partitioning via IBM Processor Resource/Systems Manager (IBM PR/SM).
 - Up to 85 Logical Partitions supported (15 each in LCSS 0 – 4, 10 in LCSS 5) for IBM z16 A01 (Machine Type 3931)
 - Up to 40 Logical Partitions supported (15 each in LCSS 0 – 4, 10 in LCSS 5) for IBM z16 A02 & AGZ (Machine Type 3932)
 - PR/SM is one of the most secure systems available, having achieved Common Criteria Evaluation Assurance Level 5+ (EAL5+) for LPAR isolation. This is one of the highest levels of certification offered that can be achieved by commercially available hardware.

Note: For more information about Common Criteria, Evaluation Assurance Levels, Protection Profiles, and a list of certified products, see [The Common Criteria](#).

The certified evaluation levels for IBM Z operating systems, as of the time of writing, are:

- ▶ IBM z/VM version 7 release 2: certified at EAL4+
- ▶ Red Hat Enterprise Linux 7.1: certified at EAL4+
- ▶ SUSE Linux Enterprise Server 15 SP2: certified at EAL4+

- The industry-leading virtualization hypervisor z/VM is supported on all IBM Z models.
- Both PR/SM and z/VM employ hardware and firmware innovations that make virtualization part of the basic fabric of the IBM Z platform.
- IBM HyperSockets allows up to 32 virtual LANs, thus allowing memory-to-memory TCP/IP communication between logical partitions (LPARs).
- ▶ Scalability
 - IBM z16 A01 scales to 200 physical processors and up to 40 TB of memory.
 - IBM z16 A02/AGZ has up to 68 physical processors and 16 TB memory
- ▶ Security:
 - The pervasive encryption capabilities of Linux on IBM Z allow you to encrypt massive amounts of data with little effect on your system performance. The Linux on IBM Z hardware benefits from encryption logic and processing on each processor chip in the system.

- The Central Processor Assist for Cryptographic Function (CPACF) is well suited for encrypting large amounts of data in real time because of its proximity to the processor unit. CPACF supports:

- DES
- TDES
- AES-128
- AES-256
- SHA-1
- SHA-2
- SHA-3
- SHAKE
- DRNG
- TRNG
- PRNG

With IBM z16, CPACF supports Elliptic Curve Cryptography clear key, improving the performance of Elliptic Curve algorithms.

The following algorithms are supported:

- EdDSA (Ed448 and Ed25519)
- ECDSA (P-256, P-384, and P-521)
- ECDH (P-256, P-384, P521, X25519, and X448)

Protected key signature creation is also supported.

- Optional cryptography accelerators provide improved performance for specialized functions:
 - Can be configured as a secure key coprocessor or for Secure Sockets Layer (SSL) acceleration.
 - Certified at FIPS 140-4 level 4.
- IBM's Hyper Protect Virtual Server offering is exclusive to Linux on IBM Z because it delivers more security capabilities to protect Linux workloads from internal and external threats throughout their lifecycle, build, management, and deployment phases. Some of the security benefits include:
 - Building images with integrity, which Secures continuous integration and delivery
 - Managing infrastructure with least privilege access to applications and data
 - Deploying images with trusted provenance
- The Linux on IBM Z maintains the Secure Execution for Linux. It is a hardware-based security technology that is designed to protect and isolate workloads on-premises, or on Linux on IBM Z hybrid cloud environments. Users, and even system administrators, cannot access sensitive data in Linux-based virtual environments.
- Regulatory compliance
 - Most security regulations feature include specific requirements regarding encryption of data and access to that data. Linux on IBM Z addresses those two aspects at the core of its platform through capabilities, such as EAL 5+ isolation between LPARs, pervasive encryption, and protection against side-channel attacks and insider threats with tamper resistant encrypted keys.
- Just-in-time deployment of resources
 - On/Off Capacity on Demand provides temporary processing capacity for meeting short-term requirements or testing new applications.
 - Flex Capacity which allows to dynamically shift production capacity between different Linux on IBM Z servers in multiple sites.

- Capacity Backup (CBU) allows you to replace model capacity or specialty engines to a backup server in the event of an unforeseen loss of server capacity because of an emergency. CBU ensures that customers can access additional capacity during a disaster recovery situation without having to purchase more capacity. Typically, this system allows customers to sign up for CBU on a Linux on IBM Z at another site and use this capacity for a number of contracted disaster recovery tests or for a contracted time during a declared disaster at the customer site. For more information about CBU, check the *IBM z16 (3931) Technical Guide*, SG24-8951.
- Power and cooling savings
 - With its low power and cooling requirements, the IBM z16 is an ideal platform for the consolidation of distributed servers.
 - Consolidating hundreds of distributed servers to IBM z16 reduces the power and cooling load in the data center.
 - The IBM Systems Director Active Energy Manager (AEM) provides a single view of actual energy usage across heterogeneous IBM platforms within a data center. AEM allows tracking of trends, which provide accurate data to help properly estimate power inputs and more accurately plan data center consolidation or modification projects.

Note: For more detailed studies about IBM Z, consult the following IBM Redbooks technical guides:

- [IBM z16 \(3931\) Technical Guide](#), SG24-8951
- [IBM z16 A02 and IBM z16 AGZ Technical Guide](#), SG24-8952
- [IBM z16 Configuration Setup](#)

1.3 Enabling a microservices architecture on Linux on IBM Z

Many organizations are trying to modernize their applications to use new approaches in the cloud. Microservices architecture is becoming the new standard for application development with more organizations moving away from traditional, monolithic applications. Breaking up large applications into a suite of smaller services allows businesses to deliver value faster, allowing their developers to work in parallel and independently.

Containers offer a convenient standard unit to encapsulate a small application component, which makes it a good infrastructure for building microservice applications. Running container-based applications on Linux on IBM Z can support consolidation efforts because multiple containers can run in single virtual machines (VMs), which allows for fewer VMs to be created because containers provide application, memory, and data isolation.

Combining the applications and all associated dependencies allows applications to be developed and compiled to run on multiple architectures, which enables application portability and provides flexibility. Through the use of containers, workloads can be deployed on any public, private, or hybrid cloud. Interoperability is essential to making it easy to move workloads from one place to another.

IBM developed its Secure Service Container (SSC) technology, which is exclusive to Linux on IBM Z, to provide an easy-to-deploy secure hosting appliance for container-based applications that run in hybrid cloud environments. SSC is a secure computing environment for microservices-based applications that can be deployed without any application code changes, which makes it an easily consumable solution for cloud-native development. It provides several unmatched security benefits, such as automatic pervasive encryption of

data in-flight and at-rest, protection from privileged administrators, and tamper protection during installation and start time to protect against malware.

An example of how a traditional application architecture and a microservices architecture can run on the same Linux on IBM Z or LinuxONE platform is shown in Figure 1-1.

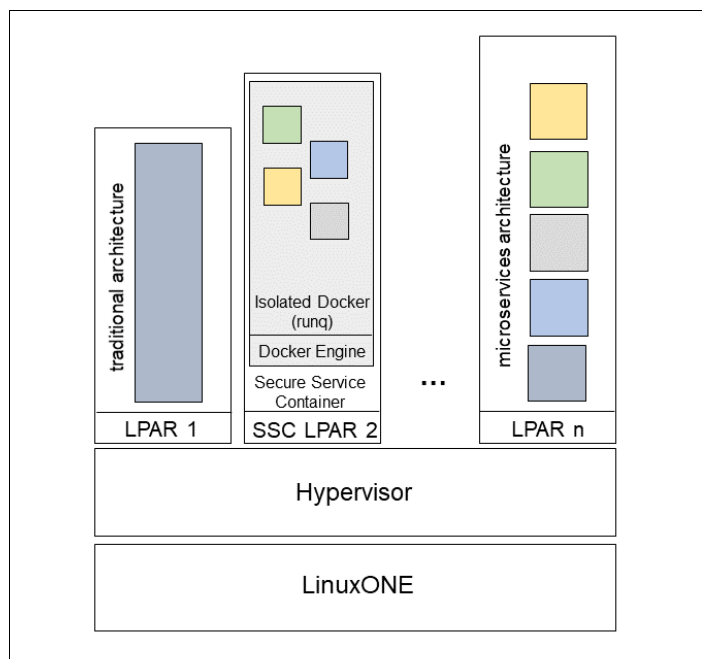


Figure 1-1 Traditional architecture, Secure Service Container, and microservices architecture

Although microservices bring many advantages, managing a larger number of services can be challenging. Orchestration tools, such as Kubernetes, can support overcoming this challenge by helping manage the deployment, placement, and lifecycle of containers.

Kubernetes is the most popular open-source container-orchestration tool; however, businesses often require extra assurance that the open-source code is safe to deploy and 24x7 support.

Red Hat OpenShift is a container-based application platform that is built on open source technologies and principles, such as Kubernetes, and enhances them and enables enterprises to also benefit from open source. Red Hat OpenShift Container Platform 4.2 was released for Linux on IBM Z servers in February of 2020. Since then, organizations can use enterprise server capabilities, such as security features, scalability, and reliability, to run cloud-native applications and accelerate their digital transformation. Application developers and operations teams can use OpenShift as a powerful tool to easily manage their container environment and see how they are organized through an intuitive web interface.

1.4 Financial benefits of a migration

In today's rapidly evolving technological landscape, the importance of secure and highly available infrastructure cannot be overstated.

As organizations navigate cost reduction initiatives, grapple with stringent privacy regulations, tackle data sovereignty challenges, Environmental, Social, and Governance (ESG) issues, especially environmental sustainability, embark on digital transformation journeys, and confront the complexities of artificial intelligence and quantum computing, the need for robust infrastructure becomes paramount.

One of the most pivotal advantages offered by Linux on IBM Z is its significant Total Cost of Ownership (TCO) optimization, which extends to various areas such as software licenses, CO2 carbon footprint reduction, and the inclusion of enhanced availability, security, and reliability features.

Businesses leverage TCO to conduct a comprehensive cost comparison between IBM solutions and alternative options, whether from competitors or in-house solutions. When constructing a TCO for their next technology investment, it's imperative to consider several key components, including:

- ▶ Full environment coverage: this encompasses all environments, from testing and development to quality assurance, staging, production, and disaster recovery (DR).
- ▶ Comprehensive component inclusion: this involves accounting for various elements such as hardware, software, cloud services, personnel, network infrastructure, security measures, storage solutions, and data center facilities.
- ▶ Business Value Assessment: evaluating the business value entails examining factors like time-to-market, customer retention rates, forecasting accuracy, scheduling efficiency, data privacy compliance, data sovereignty considerations, and potential SLA penalties.
- ▶ Quality of service metrics: this category focuses on assessing the availability, security robustness, reliability, performance levels, scalability potential, and portability of the solution.
- ▶ Time-related considerations: this involves factoring in aspects like upgrade cycles, refresh intervals, migration efforts, sunk costs, parallel costs during transition periods, and other time-sensitive factors.

IBM offers additional support through its [IT Economics team](#), which conducts Business Value Assessments (BVAs). Unlike TCO studies, BVAs emphasize return on investment (ROI) analyses. For instance, when evaluating IBM Cloud Paks on Linux on IBM Z or Red Hat OpenShift deployments, the discussion may revolve around investments, particularly concerning transfers, upgrades, or license exchanges. These discussions transcend mere technological considerations; they delve into understanding how different organizations structure their licensing models and how such insights inform decision-making based on financial benefits.

Each of the mentioned features detailed in 1.2, “Reasons to select Linux on IBM Z” on page 2 contributes significantly to reducing Total Cost of Ownership (TCO) in various ways:

- ▶ Zero trust security: by implementing a zero-trust security model and incorporating features like Secure Execution for Linux and end-to-end encryption (including quantum-safe encryption), organizations can mitigate security risks and potential breaches. This, in turn, helps avoid costly security incidents, regulatory fines, and reputational damage associated with data breaches, ultimately reducing the overall TCO.
- ▶ Scalability with ease: the on-premises cloud-like capacity model allows organizations to scale their infrastructure efficiently according to their changing needs. This scalability eliminates the need for over-provisioning resources, which can result in wasted investments. As a result, organizations can optimize resource utilization and avoid unnecessary spending, thus reducing TCO.
- ▶ Performance improvement: achieving an up to 10X performance improvement over x86 platforms translates to higher efficiency and productivity within the organization.

With faster processing speeds and improved system responsiveness, organizations can accomplish tasks more quickly, leading to increased operational efficiency and potentially lower labor costs, thereby reducing TCO.

- ▶ **Low latency and co-location:** reduced latency and co-location capabilities result in faster data transfer and processing times. This is particularly beneficial for latency-sensitive applications such as financial transactions or real-time analytics. By minimizing delays and optimizing data transfer, organizations can improve productivity and avoid potential revenue losses due to sluggish performance, thus reducing TCO.
- ▶ **High Availability:** The promise of “Seven 9s of availability” ensures minimal downtime and maximum uptime for critical business operations. This high availability minimizes the risk of revenue loss associated with system outages and service disruptions. By maintaining continuous operations, organizations can avoid costly downtime-related losses, enhancing overall cost-effectiveness and reducing TCO.
- ▶ **Software license savings and operational efficiency:** by leveraging Linux on IBM Z features, organizations can optimize their software licensing costs through efficient resource utilization and workload consolidation. Additionally, the platform's energy-efficient design reduces power and cooling requirements, leading to lower operational expenses. Furthermore, the reduced footprint saves on floor space, which can be a significant cost consideration in data center operations.
- ▶ **Reduced energy consumption and operational costs:** by leveraging a highly efficient system, organizations can significantly decrease their energy consumption and operational expenses. Such systems are designed with energy-efficient components, optimized power management features, and advanced cooling mechanisms, all of which contribute to lower energy usage. As a result, organizations can save substantially on electricity bills and reduce their overall operational costs, thus driving down the TCO. Additionally, reduced energy consumption aligns with sustainability goals, further enhancing cost savings and corporate responsibility efforts.
- ▶ **Reduction of floor space:** Linux on IBM Z is engineered to optimize space utilization in data centers. With its high consolidation ratio and support for virtualization technologies such as KVM and z/VM, Linux on IBM Z enables organizations to run multiple workloads on a single physical server, reducing the number of servers required and hence the floor space needed.
- ▶ **Consistent transactional service levels:** a massively scalable system ensures consistent transactional service levels, even during periods of high demand or peak loads. This reliability in performance helps organizations avoid disruptions, downtime, and service degradation, which can incur significant costs in terms of lost productivity, revenue, and customer satisfaction. By maintaining consistent service levels, organizations can minimize the need for costly emergency interventions, such as system upgrades or additional infrastructure investments, thereby lowering the overall TCO.

IBM LinuxONE or Linux on Z TCO and CO2e Calculator

IBM LinuxONE or Linux on Z TCO and [CO2e Calculator](#) is a tool designed to assess the environmental impact and Total Cost of Ownership (TCO) differences between x86 architecture and IBM LinuxONE or Linux on Z platforms when running similar workloads.

Key features include:

- ▶ **CO2e Emission Comparison** by evaluating the carbon dioxide equivalent (CO2e) emissions associated with running workloads on x86 versus IBM LinuxONE or Linux on IBM Z.
- ▶ **License Consolidation for TCO Savings:** the calculator demonstrates the potential TCO savings achieved by consolidating per-core licenses on IBM LinuxONE or Linux on IBM Z platforms.

- Floor space reduction: the TCO calculator evaluates the floor space gain when compared to x86 servers. By analyzing factors such as server density, power consumption, and cooling requirements, the calculator provides insights into the physical footprint reduction achievable with Linux on IBM Z.
- Cost Reduction through enterprise software: running enterprise software on IBM LinuxONE or Linux on Z can lead to significant IT cost reductions.

Figure 1-2 shows the TCO based on a current infrastructure setup, comprising of 10 rack servers with 64 cores per server running Open Source databases, 20 rack servers with 56 cores per server running Commercial databases, and an additional 20 rack servers with 56 cores per server dedicated to running commercial applications.




Age of servers	Types of servers	Workload	# of physical servers	Total cores per server	Delete
2 years old	Rack server	Open source database	10	64	
4 years old	Rack server	Commercial application	20	56	
4 years old	Rack server	Commercial database	20	56	

Figure 1-2 Existing x86 rack servers

Upon evaluation, the IBM LinuxONE or Linux on Z TCO and CO2e calculator proposes consolidating all workloads onto two IBM LinuxONE 4 Emperor systems, with a total of 296 Integrated Facility for Linux (IFL) processors as described in Figure 1-3 on page 11, showcasing the tangible benefits of migrating to the Linux on IBM Z environment.

Your x86 server inputs

Servers	Type of servers	Workload	Processors per server	Cores per x86 server	# of physical production servers	# of physical non-production servers ¹	Total DR servers ²	Total DR cores ²	Total x86 servers	Total x86 cores
2 years old	Rack	opendb	2	64	10	10	0	0	20	1280
3 years old	Rack	database	2	56	20	20	0	0	40	2240
3 years old	Rack	Application	2	56	20	20	0	0	40	2240

1. For each set of production workloads is an additional 100% of corresponding physical servers for the DevTest and Quality Assurance non-production environment. A production workload environment of 100 cores, for example, is assumed to require another 100 cores for supporting non-production DevTest and QA work.

2. The DR environment is assumed to replicate the production environment only, so corresponding non-production workloads are not included for DR.

IBM[®] LinuxONE or Linux on Z alternative

IBM [®] LinuxONE or Linux on Z model	Type of servers	Workload	# of IBM [®] LinuxONE or Linux on Z systems	# of IBM [®] LinuxONE or Linux on Z production cores	# of IBM [®] LinuxONE or Linux on Z non-production cores ³	Total DR servers ²	Total DR cores ²	Total IBM [®] LinuxONE or Linux on Z servers	Total IBM [®] LinuxONE or Linux on Z cores ⁴
LinuxONE 4 Emperor	19-inch frame	opendb, database, Application	2	169	127	0	0	2	296

1. For each set of production workloads are an additional 75% corresponding DevTest and Quality Assurance non-production environments that can reside within the same physical IBM[®] LinuxONE or Linux on Z server. A production workload environment of 10 cores, for example, is assumed to require another 7.5 cores for supporting non-production DevTest and QA work.

2. The DR environment is assumed to replicate the production environment only, so corresponding non-production workloads are not included for DR.

3. Total required IBM[®] LinuxONE or Linux on Z cores are rounded up to the next whole number of cores.

Figure 1-3 Linux on IBM Z sizing based on x86 inputs

This consolidation yields remarkable benefits as depicted in Figure 1-4 on page 12.

- **58% Lower Energy Consumption:** by transitioning to the Linux on IBM Z infrastructure, energy consumption is significantly reduced compared to the existing setup, leading to substantial cost savings and environmental benefits as depicted in Figure 1-4 on page 12.

CO₂ Emissions

x86 CO₂ Emissions

	Year 1	Year 2	Year 3	Year 4	Year 5	Total
Yearly kWh	1,069,449	1,069,449	1,069,449	1,069,449	1,069,449	5,347,244
Emissions (CO ₂ Eqv)	381,077	381,077	381,077	381,077	381,077	1,905,384

IBM® LinuxONE or Linux on Z CO₂ Emissions

	Year 1	Year 2	Year 3	Year 4	Year 5	Total
Yearly kWh	453,856	453,856	453,856	453,856	453,856	2,269,278
Emissions (CO ₂ Eqv)	161,722	161,722	161,722	161,722	161,722	808,612

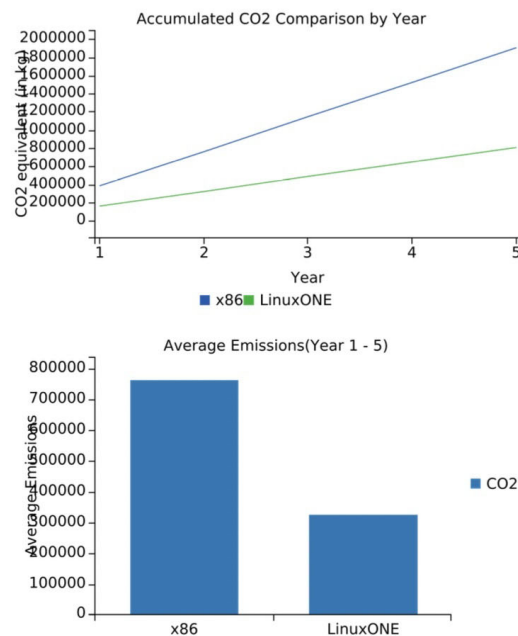


Figure 1-4 Co2 emissions Comparison

- **33% Less Floor Space:** the consolidation onto the Linux on IBM Z allows for a more efficient use of physical space, requiring less floor area for housing the infrastructure, thereby optimizing data center real estate as depicted in Figure 1-5 on page 13².

² Floor space cost is estimated to be \$10,000 per rack per year.

x86 Costs By Year

	Year 1	Year 2	Year 3	Year 4	Year 5	5 Year Total Cost
HW purchase	\$0	\$0	\$0	\$0	\$0	\$0
HW maintenance	\$229,530	\$229,530	\$229,530	\$229,530	\$229,530	\$1,147,650
Server migration	\$0	\$0	\$0	\$0	\$0	\$0
Application/Database SW	\$8,611,280	\$8,611,280	\$8,611,280	\$8,611,280	\$8,611,280	\$43,056,400
Linux OS SW	\$220,960	\$220,960	\$220,960	\$220,960	\$220,960	\$1,104,800
Systems Management and Virtualization SW	\$860,800	\$860,800	\$860,800	\$860,800	\$860,800	\$4,304,000
Electricity	\$112,394	\$112,394	\$112,394	\$112,394	\$112,394	\$561,972
Space	\$90,000	\$90,000	\$90,000	\$90,000	\$90,000	\$450,000
Labor	\$400,000	\$400,000	\$400,000	\$400,000	\$400,000	\$2,000,000
Totals	\$10,524,964	\$10,524,964	\$10,524,964	\$10,524,964	\$10,524,964	\$52,624,822

IBM® LinuxONE or Linux on Z Costs By Year

	Year 1	Year 2	Year 3	Year 4	Year 5	5 Year Total Cost
HW purchase	\$13,047,807	\$0	\$0	\$0	\$0	\$13,047,807
HW maintenance	\$0	\$0	\$0	\$1,557,010	\$1,557,010	\$3,114,020
Server migration	\$1,000,000	\$0	\$0	\$0	\$0	\$1,000,000
Application/Database SW	\$604,777	\$604,777	\$604,777	\$604,777	\$604,777	\$3,023,887
Linux OS SW	\$1,618,582	\$1,618,582	\$1,618,582	\$1,618,582	\$1,618,582	\$8,092,909
Systems Management and Virtualization SW	\$0	\$0	\$0	\$302,382	\$302,382	\$604,764
Electricity	\$47,698	\$47,698	\$47,698	\$47,698	\$47,698	\$238,491
Space	\$60,000	\$60,000	\$60,000	\$60,000	\$60,000	\$300,000
Labor	\$200,000	\$200,000	\$200,000	\$200,000	\$200,000	\$1,000,000
Totals	\$16,578,865	\$2,531,057	\$2,531,057	\$4,390,450	\$4,390,450	\$30,421,879

Figure 1-5 Breakdown of x86 costs vs. IBM Linux on IBM Z costs by year

- ▶ 76% Lower Software Costs: through consolidation and the inherent efficiencies of Linux on IBM Z, software costs are substantially reduced, contributing to significant savings over the long term, as shown in Figure 1-6.

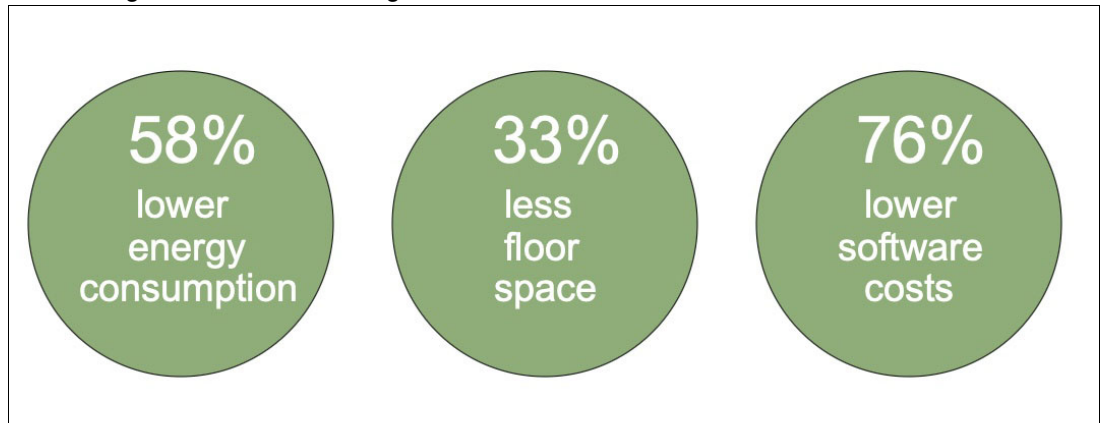


Figure 1-6 Linux on IBM Z tangible benefits

Additionally, Figure 1-7 provides a comprehensive overview of the 5-Year Category Cost Comparison Highlights, highlighting the substantial reductions across various cost categories.

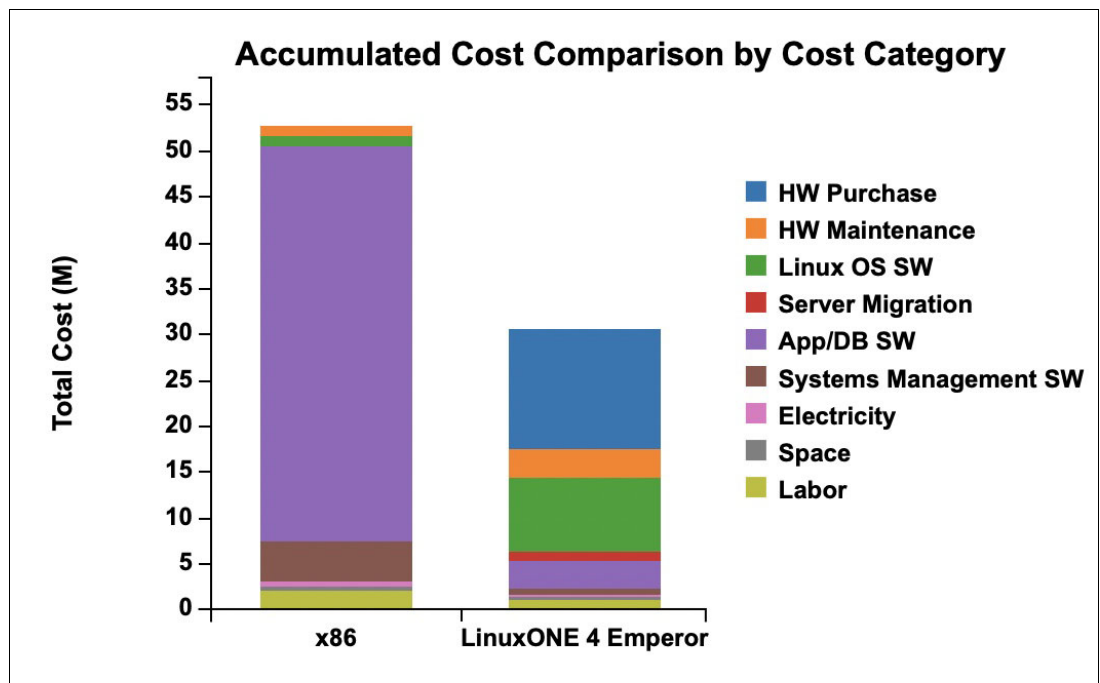


Figure 1-7 Accumulated Cost Comparison by Cost Category

1.5 Cloud computing blueprint for Linux on IBM Z

Linux on IBM Z is designed to play an integral part in your private, public or hybrid cloud infrastructure. A wide range of organizations, from small businesses to large institutions, are using the unique security capabilities of Linux on IBM Z to power their private and hybrid cloud. Cloud Service Providers are using Linux on IBM Z technology to add next-level security and stability to its public cloud services. Linux on IBM Z also plays a critical role in IBM's public cloud offering as the Hyper Protect Service Portfolio is based on the technology.

1.5.1 Cloud Solutions on Linux on IBM Z

Mission-critical applications are the backbone of an organization. Any downtime has a major impact to a business. These workloads require a cloud solution that has exceptional system uptime, excellent data security and privacy, and a powerful vertical scale architecture. For these workloads, Linux on IBM Z is the most secure, reliable, and scalable on-premises cloud solution.

Linux on IBM Z can provide the same agility and time to value as other cloud services, along with unparalleled enterprise qualities of service. Linux on IBM Z allows those delivering cloud services to rapidly deploy a trusted, scalable OpenStack-based Linux cloud environment that can start small and scale up massively to support thousands of virtual servers or up to two million containers on a single system.

Virtualization portfolio

Establishing cloud environments on Linux on IBM Z begins with virtualization technology. Customers have a choice of deploying z/VM, the world's first commercially available hypervisor to provide virtualization technology, or the newer industry-standard KVM. Both hypervisors allow you to bring new virtual servers online in a matter of minutes (or less) to accommodate growth in users, although each technology is designed with a different audience in mind.

The overall IBM virtualization portfolio includes the following applications for infrastructure and virtualization management:

- ▶ Linux on IBM Z Hardware: IBM z16:
 - Massively scalable.
 - Characterized by excellent economics and efficiencies.
 - Highly secure and available.
- ▶ z/VM:
 - Support more virtual servers than any other platform in a single footprint.
 - OpenStack support.
- ▶ KVM for Linux on IBM Z:
 - Provides a choice for clients who want open virtualization while taking advantage of the robustness, scalability, and security of the Linux on IBM Z platform.
 - The standard interfaces that it provides allows for easy integration into an existing infrastructure.
- ▶ Linux on IBM Z:

Distributions are available from Red Hat Enterprise Linux, SUSE Linux Enterprise Servers, and Ubuntu.

- ▶ IBM Wave for z/VM:

A graphical interface tool that simplifies the management and administration of a z/VM and Linux environment.

Note: IBM Wave for z/VM has been withdrawn from marketing:

- ▶ [31 March 2022, for Wave standalone product \(5648-AE1\)](#)
- ▶ 30 September 2022, for Wave as a component of IISz V1 (5698-IS2)

Consider Cloud Infrastructure Center as an alternative. Customers requiring additional entitlement or continued support and enhancements should pursue an agreement with [Log-on Software](#).

- ▶ IBM Dynamic Partition Manager (DPM):

Tool for Linux on IBM Z configuration and setup to simplify and speed up deployment of Linux servers by using only the Hardware Management Console (HMC).

Cloud Solutions on Linux on IBM Z

To provide cloud management capability, z/VM and KVM are OpenStack-enabled, which is the industry standard for ubiquitous cloud computing platforms. Applications that use the OpenStack application programming interfaces (APIs) are supported on both hypervisors.

IBM Cloud Infrastructure Center is an advanced infrastructure management offering, including on-premises cloud deployments of IBM z/VM-based and KVM-based Linux virtual machines on Linux on IBM Z. It is an industry-proven turn-key Infrastructure-as-a-Service (IaaS) solution that provides a consistent, industry-standard user experience to define, instantiate, and manage the lifecycle of virtual infrastructure, deployment of images (operating system and applications), and policies to maximize resource utilization.

For more information about how to install, configure, and use the IBM Cloud Infrastructure Center, see the IBM Cloud Infrastructure Center documentation, which is available at [IBM Knowledge Center](#).

IBM Cloud Infrastructure Center delivers the following capabilities:

- ▶ Easy provisioning of virtual machine instances into an on-premises cloud by way of a self-service portal.
- ▶ Infrastructure provisioning that can be confined by workflow-driven policies.
- ▶ Automated configuration of I/O and network resources.
- ▶ Image management that includes virtual machine image capture, catalog, and deployment.
- ▶ Easy integration into higher-level cloud automation and orchestration tools.
- ▶ Federation of an on-premises cloud with OpenStack clouds by way of OpenStack compatible APIs that establish a multi-region cloud.
- ▶ Manage existed virtual machines that are not created by compute nodes with onboarding feature (z/VM only).

The IBM Cloud Infrastructure Center architecture on z/VM is shown in Figure 1-8.

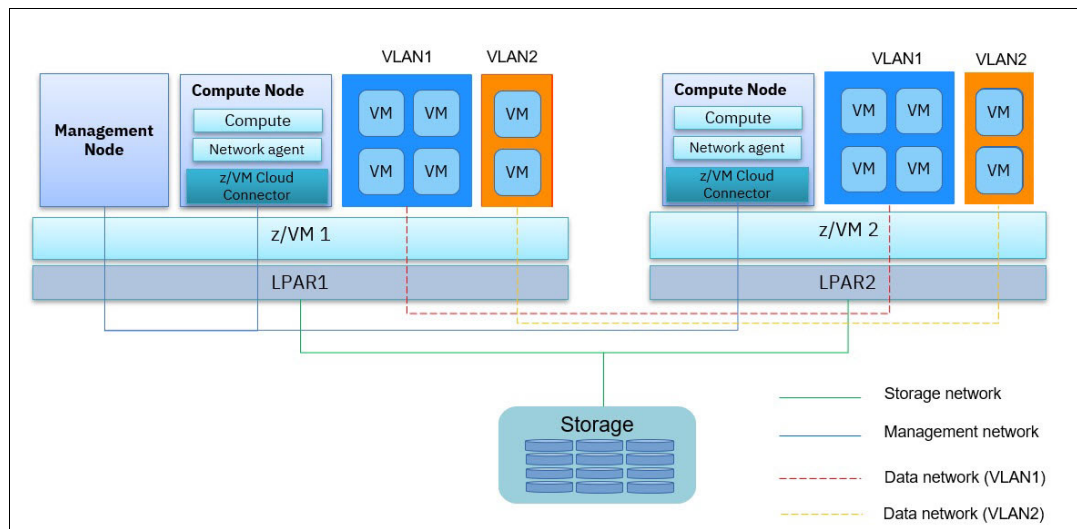


Figure 1-8 IBM Cloud Infrastructure Center Architecture on z/VM

Consider the following points regarding the IBM Cloud Infrastructure Center Architecture on z/VM that is shown in Figure 1-8:

- ▶ Only one management node must be set up for managing the entire z/VM Cloud infrastructure.
- ▶ For each to-be-managed z/VM, one compute node is required.
- ▶ The management node can be in the same z/VM instance with one of the compute nodes, but they must be on different Linux virtual machines of that z/VM.

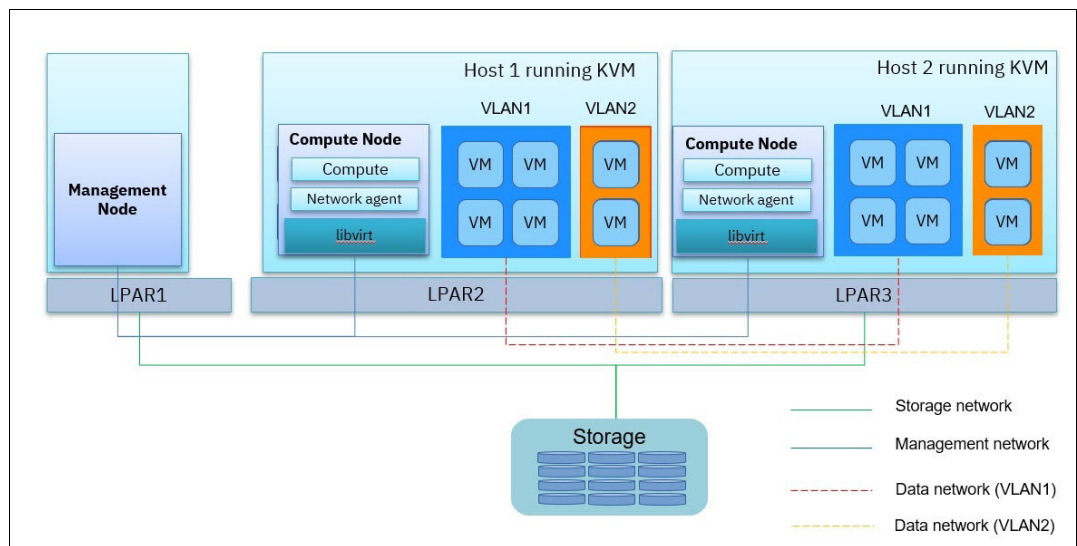


Figure 1-9 IBM Cloud Infrastructure Center Architecture on KVM

Consider the following points regarding the IBM Cloud Infrastructure Center Architecture on KVM that is shown in Figure 1-9:

- ▶ Only one management node must be set up for managing the entire KVM Cloud infrastructure.
- ▶ For each to-be-managed KVM, one compute node is required.

- The management node can be on the same KVM instance with one of the compute nodes, but it is highly recommend to separate management node and compute node into different LPAR.

Other industry OpenStack-based cloud management solutions can also run on Linux on IBM Z, including (but not limited to) the VMware vRealize Automation product.

Red Hat OpenShift on Linux on IBM Z

IBM announced the availability of Red Hat OpenShift on Linux on IBM Z in 2020, making the integration of applications running on Linux on IBM Z with applications running elsewhere in the cloud much easier. OpenShift supports cloud-native applications being built once and deployed anywhere. Application developers and operations teams use OpenShift to easily and efficiently manage their container environment.

The minimum OpenShift architecture consists of five Linux guests (bootstrap, three control nodes, and one worker node) that are deployed on top of IBM z/VM 7.1. Customers that use OpenShift on Linux on IBM Z can use the IBM Cloud Infrastructure Center to manage the underlying cluster infrastructure.

1.6 Linux on IBM Z flexibility and sustainability

With a long history of environmental achievement at IBM and a focus on environmental issues across all aspects of the product development, IBM Z is the ideal platform for achieving your sustainability goals.

In 2018, IBM introduced the IBM z14® ZR1. With the IBM z14 ZR1, IBM introduced a new standardized form factor designed for use in the modern cloud data center. The new system design has changed the footprint for IBM Z servers. IBM z16, like its predecessor, is built with a 19" frame that flexibly scales from 1 – 4 frames depending on the configuration.

This means that for many clients they will actually see a reduction in the amount of floor space taken up within the data center, but perhaps more importantly this system can now fit within your standard data center aisles – offering new options for hot aisle / cold aisle configurations in a data center. With the new option of intelligent Power Distribution Unit (iPDU), you can fit even more I/O capacity without needing extra frames, save power, and move to a standardized approach within your data center.

The IBM Z platform offers differentiated architectural advantages including:

- Better per core performance than x86.
- Designed for seven 9's (99.99999%) availability with new system recovery boost enhancements to accelerate middleware restart and diagnostic processing and parallel sysplex scale and performance.
- Capacity backup units allow capacity to be added to the system, without requiring additional hardware to be deployed in the data center.
- The ability to sustain high CPU utilization and minimize wasted resources.
- With the introduction of the new iPDU power option (2N internal, fully integrated Intelligent Power Distribution Units), a typical IBM Z configuration consume less power than a comparably configured previous generation of IBM Z (similar number of processors, number/type of I/O cards, amount of memory).

- ▶ In addition to the energy savings, the modularity and scalability of the platform, now available from 1 to 4 19-inch frames, also enables large improvements in data center space and power efficiency.
- ▶ The new form factor enables direct participation in the most energy-efficient data center cooling technologies, such as hot and cold air containment cooling systems.
- ▶ IBM z16 supports [ASHRAE](#) Thermal Guidelines for Data Processing Environments 4th edition, which reduces the requirements for humidification and its energy consumption.
- ▶ The IBM z16 also marks a distinct sustainability focus across the product lifecycle, from the improved energy efficiency, enhancement of manufacturing and material sourcing, to the improved packaging strategies for shipment, to material recycling at product end-of-life.



Analyze and understand

This chapter outlines some of the points that you need to consider before deciding to migrate to the Linux on IBM Z platform. This chapter also provides a description of total cost of ownership (TCO) and helps you to analyze which workloads would make good candidates for migration. Also, we touch on the financial benefits of migration.

This chapter includes the following topics:

- ▶ 2.1, “Total cost of ownership analysis” on page 22
- ▶ 2.2, “Choosing workloads to migrate” on page 23
- ▶ 2.3, “Financial benefits of a migration” on page 24
- ▶ 2.4, “Overview of migration steps” on page 26
- ▶ 2.5, “Analysis of how to size workloads for migration” on page 27

2.1 Total cost of ownership analysis

Many CIOs recognize the return on investment (ROI) in the information technology of their companies, but at the same time they are frustrated by an increasingly costly IT infrastructure. There are many reasons for these costs, some of which are the annual costs of software licensing, power, cooling, and ongoing support. The complexity of environments usually determines the magnitude of these costs.

Traditional TCO calculations often focus on readily quantifiable costs:

- ▶ Direct costs: initial purchase price, hardware and middleware expenses, licensing fees, maintenance, and support contracts.
- ▶ Indirect costs: training, downtime due to outages, productivity losses, potential opportunity costs and sunk costs.

Failing to integrate security, privacy, and sustainability into your TCO analysis leads to an incomplete picture of potential risks and long-term costs, such as:

- ▶ Security
 - Compromised systems, stolen data, and network infiltration carry massive consequences. These include direct remediation costs, legal fines, reputational damage, and lost business opportunities.
 - Proactive security measures like vulnerability assessments, penetration testing, employee training, and robust security systems reduce exposure but are often overlooked in basic TCO calculations.
 - Including security as a foundational pillar in TCO reveals its true value as an investment that protects against potentially crippling financial losses.
- ▶ Privacy
 - Stringent regulations such as the General Data Protection Regulations (GDPR) mandate strict adherence to data protection principles. Failure to comply results in substantial fines and legal repercussions.
 - Consumers are increasingly privacy-conscious. Mishandling personal data can seriously damage an organization's reputation, leading to erosion of customer trust and market share.
 - Investing in privacy controls, data governance mechanisms, and secure data handling practices demonstrates good corporate citizenship and mitigates long-term risks.
- ▶ Sustainability
 - Technology has a significant environmental footprint. E-waste, energy consumption, and carbon emissions contribute to ecological damage and climate change.
 - Energy-efficient hardware, environmentally conscious disposal practices, and sustainable procurement strategies may incur upfront costs but reduce long-term operational expenses.
 - Consumer and stakeholder expectations are shifting toward ethical and sustainable business practices. Incorporating sustainability initiatives can enhance brand perception and align with corporate values.

A holistic TCO calculation that integrates security, privacy, and sustainability paints a far more accurate picture of an asset's true lifetime cost.

A large North American client operates in a cloud native MongoDB-as-a-Service environment, prioritizing regulatory compliance and sustainability. Their business necessitates Federal Financial Institutions Examination Council (FFIEC) Appendix J-compliant MongoDB solutions for critical applications, helping ensure cyber resilience and sub-second recovery for multi-terabyte instances. With a core consolidation ratio of 33:1 compared to x86, the solution on IBM Z architecture offers high performance and availability. Key components include Mongo Enterprise on Linux on IBM Z, data encryption, FFIEC Appendix J compliance, and Mongo Operations Manager for logging and monitoring.

For more case studies, see [Case Studies](#).

2.2 Choosing workloads to migrate

When you have decided to migrate and consolidate, the next step is to examine which workloads would be good candidates to be migrated.

Several variables must be considered, such as:

- ▶ Associated costs
- ▶ Application complexity
- ▶ Service-level agreements (SLAs)
- ▶ Skills and abilities of your support staff

Start with a not overly complex application that has a low SLA and a staff that has the associated skills.

In the case of home-grown applications, ensure that you have the source code available. Regarding the operating system platform, even a workload from a different platform can be migrated but start with servers running Linux.

IBM Z has a powerful processor with a clock speed of 5.2 GHz (z16), up to 200 cores and 40 TB of memory. Because IBM Z is designed to concurrently run disparate workloads, it is important to remember that some workloads that required dedicated physical processors that are designed to run at high sustained CPU utilization rates may not be optimal candidates for migration to Linux on IBM Z. This is because workloads that require dedicated processors will not take advantage of the virtualization and hardware sharing capabilities. An example of such an application might include video rendering, which requires specialized video hardware.

2.2.1 Ideal use cases for Linux on IBM Z

Workloads suitable for consolidation on Linux on IBM Z include:

- ▶ Database servers: Databases like MongoDB, EnterpriseDB, and PostgreSQL can benefit from the high-performance architecture of IBM Z, especially when handling large volumes of data.
- ▶ Web servers: Applications requiring high availability and reliability, such as web servers running Apache or Nginx, can be consolidated on IBM Z to optimize resource utilization.
- ▶ Virtualization: Linux on IBM Z provides robust virtualization capabilities, making it suitable for consolidating virtualized workloads, including development and testing environments.
- ▶ Analytics and Big Data: Workloads involving analytics, data processing, and big data applications can leverage the scalability and performance of IBM Z for efficient processing.

- ▶ DevOps and CI/CD: Tools and pipelines for continuous integration and continuous delivery (CI/CD) can be consolidated on IBM Z to streamline software development processes.
- ▶ Artificial intelligence (AI) and machine learning (ML): Workloads related to AI and ML can benefit from the high computational power and scalability offered by IBM Z.

2.2.2 Ideal use cases for Red Hat OpenShift on IBM Z

Workloads suitable for consolidation on Red Hat OpenShift on IBM Z include:

- ▶ Data gravity: Red Hat OpenShift colocated with IBM Z leverages existing data on the platform, reducing latency, and maximizing data value. Whether it is IBM CICS®, Db2, or IMS, being next to the data sources accelerates application deployment and versioning, facilitating faster time to market. Additionally, the coexistence of IBM z/OS® and Red Hat OpenShift offers flexibility and efficiency.
- ▶ Consolidation and TCO reduction: Red Hat OpenShift on IBM Z and LinuxONE brings economic and operational advantages by requiring fewer physical resources, reducing the number of operational endpoints to manage, and enabling dynamic workload handling without additional hardware footprint.
- ▶ Business continuity: IBM Z's reputation for business continuity, along with pervasive encryption capabilities, ensures data security and reliability. High scalability, predictable latency between LPARs, and dynamic resource allocation based on service level agreements (SLAs) further enhance business continuity measures.
- ▶ **Blockchain and Digital Assets Management:** Beyond cryptocurrency, blockchain secures digital assets and transactions. IBM Z Crypto Cards, HSM, and Hyper Protect Virtual Servers ensure data and environment security. The flexibility of Red Hat OpenShift enables deployment on secure platforms, facilitating collaboration while maintaining security.
- ▶ **Cross-platform application development consistency:** Red Hat OpenShift enables seamless deployment of containers across different platforms like IBM Z and LinuxONE. Using multi-architecture manifests, containers can be deployed on various platforms without modification, streamlining deployment through automation and CI/CD pipelines.

Chapter 6, “Migration analysis” on page 71, provides an in-depth analysis of the process of determining the most appropriate applications to migrate to a Linux on IBM Z environment.

2.3 Financial benefits of a migration

Additional benefits of IBM Z include cost reduction through environment sharing. These benefits include:

- ▶ Reduced risk of downtime due to redundancy of the hardware and z/VM features like single system image (SSI) and live guest relocation. Migrating to Linux on IBM Z offers long-term cost predictability, as it reduces the risk of unexpected expenses associated with hardware failures, maintenance, and upgrades. With the reputation of IBM Z for reliability and stability, organizations can better forecast and manage their IT budgets over time.
- ▶ Savings on software licensing: Databases, operational systems, application server, and management software in a current distributed server farm can be licensed more cost effectively using the specialized IBM Z processor Integrated Facility for Linux (IFL).
- ▶ Improved scalability and flexibility: Linux on IBM Z provides scalability and flexibility that can adapt to changing business needs without requiring significant infrastructure

investments. Organizations can easily scale up or down based on demand, optimizing resource usage and minimizing costs associated with over-provisioning or under-utilization.

- **Save energy and be green:** When you have hundreds or thousands of servers consolidated in a single box, the energy and cooling costs can be reduced up to 75% in comparison to a distributed environment.
- **Enhanced security and compliance:** Built-in security features, such as pervasive encryption and hardware-based cryptographic capabilities, help organizations strengthen their security posture and achieve compliance with industry regulations. Avoiding security breaches and non-compliance penalties can result in significant cost savings and reputational benefits.
- **Save costs of ongoing support:** The complexity of maintenance of the environment is decreased since you have many virtual servers in a single box.

Figure 2-1 shows an up to 82% software cost reduction in comparison with x86 servers when using the [Linux on IBM Z TCO and CO2e Calculator](#).

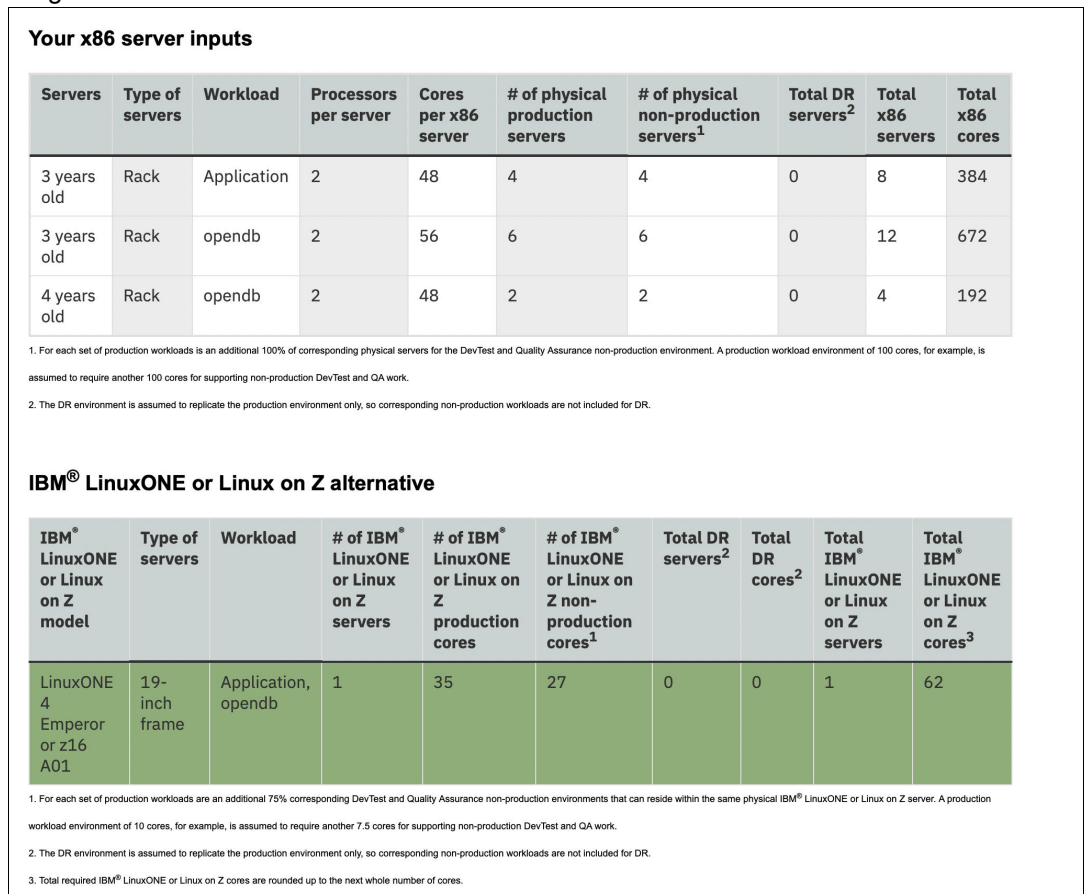


Figure 2-1 Software cost reduction according to IBM LinuxONE or Linux on IBM Z TCO calculator

The cost savings arise because Linux on IBM Z is treated by most software vendors as a distributed system, and software is usually charged by the core. Because an IFL is classified as a single core, and has high processing power, there could be significant savings by consolidating multiple distributed servers to an IFL. Figure 2-2 on page 26 shows an example company that has 45 virtual servers and uses only 14 licenses.

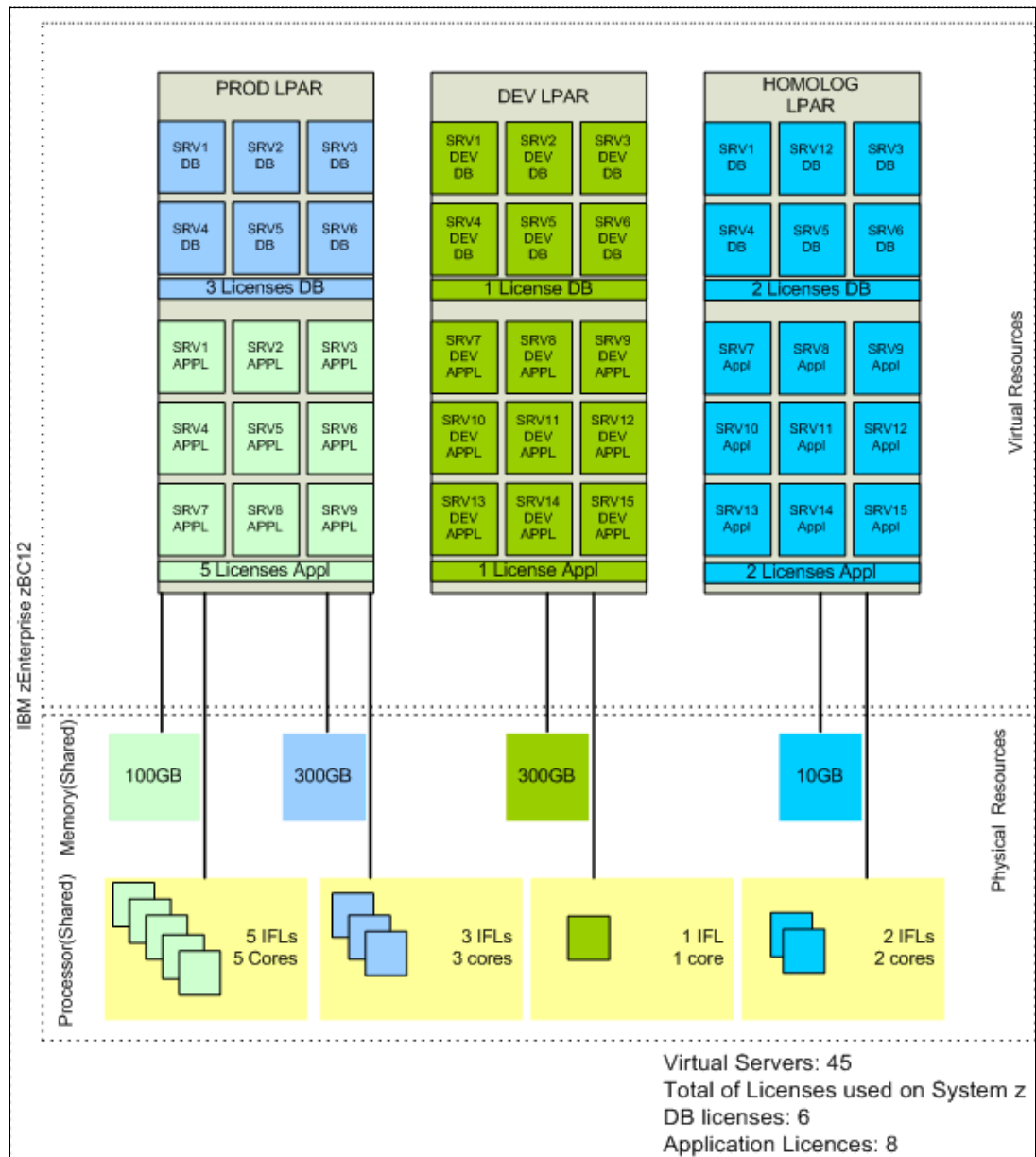


Figure 2-2 Example company saving licenses

Note: For an accurate TCO study, contact your software vendor or IBM representative to understand its policies and pricing regarding application consolidation on Linux on IBM Z.

2.4 Overview of migration steps

The migration process requires several steps, and anticipating how much time is needed for each step is crucial.

The following are the phases of migration at the highest level:

- ▶ **Planning and assessment:** After you have decided what will be migrated and how, the plan must specify the time, risks, and owner for each migration task and may include:
 - Goals and scope definition
 - Inventory and assessment
 - Feasibility study
 - Target environment design
- ▶ **PoC and Test:** Proof of concept to check the compatibilities between the x86 and Linux on the IBM Z environment, and give special focus to performance.
- ▶ **Education:** The technical staff needs the correct skills to work on Linux on IBM Z migration and maintain the new environment.
- ▶ **Build Environment:** In this phase, the new infrastructure is readied for migration.
- ▶ **Implementation:** Communication between stakeholders is important during this process. All involved people must know and approve the migration, and receive follow-up reporting on the progress. The migration execution may include:
 - Pilot migration
 - Phased migration
 - Data migration
 - Application deployment and configuration
- ▶ **Postmigration:** After implementation, documentation must be created that further references the project and documents all necessary maintenance and care procedures. Additionally, the project manager must have signed an acceptance agreement. The postmigration may include:
 - Validation and testing
 - Performance monitoring
 - Optimization and fine-tuning
 - Decommissioning of old systems
- ▶ **Additional Considerations:**
 - Develop a rollback plan in case of unforeseen issues during the migration process.
 - Ensure data security throughout the migration process and implement appropriate security measures in the new Linux on IBM Z environment.
 - You could utilize support available from IBM and other vendors involved in the migration process

In general, it can be presumed that the proof of concept, associated testing, and final implementations are the most time-consuming of a migration.

2.5 Analysis of how to size workloads for migration

One of the challenges of migration is to determine the resources that are required on the target platform to accommodate the distributed workload.

The first step is to determine the expected consolidation ratio for a specific workload type. This step allows you to answer the question “What is the theoretical maximum number of servers that can be consolidated?”

Sizing the theoretical maximum number of servers that can be consolidated requires considering a combination of factors, both technical and non-technical. Here are some key aspects to consider:

- ▶ Technical factors:
 - Server resources (as shown in Table 2-1):
 - CPU: The combined processing power of the target server (where workloads are consolidated) must be sufficient to handle the aggregate workload of the servers being consolidated, this includes processor speed (or speeds) of the servers to be consolidated and average of CPU utilization of these servers.
 - Memory: The target server needs adequate memory capacity to accommodate the combined memory requirements of all consolidated workloads.
 - Storage: Storage capacity and performance of the target server should be sufficient to store and access the data from all consolidated servers efficiently.
 - Network bandwidth: The network infrastructure must have enough bandwidth to handle the increased traffic generated by consolidated workloads.

Table 2-1 Server resource sizing example

Server Resources	
Description	Example
CPU	▶ Individual Servers: <ul style="list-style-type: none"> – Server A - 4 cores, 2.5 GHz – Server B - 2 cores, 3 GHz – Server C- 2 cores, 3 GHz ▶ Target Server: 16 cores, 5.0 GHz
Memory	▶ Individual Servers: <ul style="list-style-type: none"> – Server A - 16 GB – Server B - 8 GB – Server C- 32 GB ▶ Target Server: 64 GB
Storage	▶ Individual Servers: <ul style="list-style-type: none"> – Server A - 2 TB HDD, SATA – Server B - 1 TB SSD, NVMe – Server C- 2 TB SSD, NVMe ▶ Target Server: 10 TB SSD, NVMe
Network Bandwidth	▶ Individual Servers: <ul style="list-style-type: none"> – Server A - 1 Gbps – Server B - 1 Gbps – Server C- 10 Gbps ▶ Required Bandwidth: To be determined based on application traffic analysis.

- Workload characteristics (as demonstrated in Table 2-2 on page 29):
 - Analyze the average and peak resource usage of individual servers to understand their processing, memory, and storage demands.
 - Ensure the target server supports the operating systems, applications, and middleware dependencies of the servers being consolidated.

- Different applications have varying performance needs. Analyze the required performance levels (for example, response times, bandwidth, SLA) for each application to ensure consolidation doesn't negatively impact user experience.
- If using virtualization, consider the capabilities of the chosen virtualization platform and its resource overhead when calculating capacity.

Table 2-2 Workload characteristics

Workload characteristics	
Description	Example
Resource Usage	<ul style="list-style-type: none"> ▶ Server A: <ul style="list-style-type: none"> – CPU - 30% average, 60% peak – Memory - 80% average, 90% peak ▶ Server B: <ul style="list-style-type: none"> – CPU - 20% average, 40% peak – Memory - 50% average, 70% peak ▶ Server C: <ul style="list-style-type: none"> – CPU - 40% average, 50% peak – Memory - 50% average, 60% peak
Compatibility	<ul style="list-style-type: none"> ▶ Server A: Linux Ubuntu 20.04, Application X ▶ Server B: Linux Ubuntu 20.04, Application Y. ▶ Server C: Linux Red Hat Enterprise Linux 8, Application Z. ▶ Target Server: Needs to support Linux distributions, and Applications X, Y, and Z.
Performance Requirements	<ul style="list-style-type: none"> ▶ Application X: Requires sub-second response time for user interactions (less than 100 ms) ▶ Application Y: Can tolerate slightly higher response times (up to 500 ms). ▶ Application Z: Can tolerate slightly higher response times (up to 2 seconds). ▶ Target server must provide sufficient resources to meet these requirements for all applications.
Overhead: Consider resource overhead introduced by the chosen virtualization platform	<ul style="list-style-type: none"> ▶ Virtualization platform overhead: 5% CPU, 10% memory. ▶ Adjust resource calculations for the target server to account for this overhead.

- ▶ Nontechnical factors:
 - Organizations need to balance potential gains from consolidation with potential risks. Consolidating too many servers can increase the impact of a single point of failure.

- Increased server consolidation can make management and troubleshooting more complex. Assess the organization's IT staff capabilities and available tools to manage larger consolidated environments.
- Consider the organization's anticipated future growth when estimating the server capacity. You do not want to reach full capacity immediately after consolidation.
- Review software licensing terms to ensure they allow for consolidation and avoid potential licensing violations.

The theoretical maximum number of servers that can be consolidated does not necessarily translate to the optimal number. A thorough analysis of all these factors, along with real-world testing and performance monitoring, helps determine a practical and sustainable consolidation strategy.

Important: Others factors must be considered to get a complete TCO such as floor space, energy savings, scalability, security, and outages. For a more accurate sizing study, contact your IBM representative.



Virtualization concepts

Virtualization is a highly prized capability in the modern computing environment. Virtualization on IBM Z offers industry-leading and large-scale proven Cloud and IT optimization capabilities to drive down the costs of managing and maintaining the tremendous proliferation of servers in today's technology infrastructures.

This chapter provides helpful information about virtualization, particularly to compare and contrast the virtualization concepts of IBM mainframe computing with those commonly used by x86 distributed systems. The two have many concepts in common, yet other concepts are very different. This brief comparison provides terminology, vocabulary, and diagrams that prove helpful in planning to migrate workloads to IBM Z.

This chapter includes the following topics:

- 3.1, "The demand for virtualization" on page 32
- 3.2, "Typical x86 virtualization" on page 32
- 3.3, "IBM Z virtualization options" on page 33
- 3.4, "Single system image and live guest relocation" on page 39
- 3.5, "z/VM hypervisor components" on page 41
- 3.6, "Virtualized resources" on page 43

3.1 The demand for virtualization

As the computing environment grows in size and complexity, the sprawling infrastructure becomes more difficult to manage. As more physical servers are added to the environment, the resources, such as CPU, RAM memory, and disk are too easily wasted and cannot be efficiently used.

Virtualization turns physical hardware into logical resources that can be shared, shifted, and reused. One of the most highly prized features of virtualization is dynamically sharing or dedicating more virtual resources, such as CPU, RAM, and disk, to a virtual guest while the virtual guest is running. This process greatly eases the system administration tasks of scaling the supply of services to meet demand.

Virtualization allows a single physical server to host numerous logical servers. The servers share the physical resources to allow all the guests' servers to accomplish more than the single physical server can on its own, while maximizing the effective use of the physical resources. In such a virtual environment, the physical server is commonly called the “host” system and the logical servers are known as “guests.” Although software solutions in the industry use variations of these terms, this publication uses the terms “host” and “guest” as defined above.

Systems administrators rely on virtualization to ease and facilitate the complex work of managing increasingly complex environments. IT managers look to virtualization to address the ever-increasing demand for more computing power from customers while accommodating shrinking IT budgets.

The growing number of physical servers also increases the amount of electric power and air conditioning that is consumed in the data center. Virtualization helps to reduce the amount of electricity that is used, which reduces costs. The aspirations of a “green” data center can similarly be met in part by using virtualization.

Virtualization is widely adopted across various industries and organizations. The rise of cloud computing heavily relies on virtualization technologies to provide on-demand infrastructure and services. This further highlights its significance in the modern IT landscape.

Despite its more recent hype, virtualization has existed in advanced computing systems for quite some time. The conception of virtualization began in the late 1960s as IBM introduced the Control Program (CP)-67. This innovation quickly grew to become a defining feature of IBM hardware, including all IBM Z systems.

For more information, see [What is virtualization?](#).

3.2 Typical x86 virtualization

Briefly stated, virtualization allows a single physical server to host numerous logical servers, sharing the physical resources in such a way as to allow all the logical servers to accomplish more than the single physical server could on its own, while maximizing the effective use of the physical resources. In such a virtual environment, the physical server is commonly called the “host” system while the several logical servers are known as “guests.” Although there are several software solutions in the industry that use variations of these terms, this publication will simply use the terms “host” and “guest” as defined above.

Figure 3-1 on page 33 shows a very simple but typical way that systems administrators set up virtual services in a distributed x86 environment. A physical server employs virtualization

software (such as KVM or XEN) to install and run a Linux guest. Figure 3-1 displays a physical server (host name “x86host1”) with three separate virtual Linux guest operating systems contained on this physical host. The physical server has a fixed amount of CPU, RAM, as well as physical access to Disk and Network resources. The virtual guests are allocated CPU, RAM and Disk resources as a subset of what is available from the physical server, and the network resources are all equally shared by the guests and physical host.

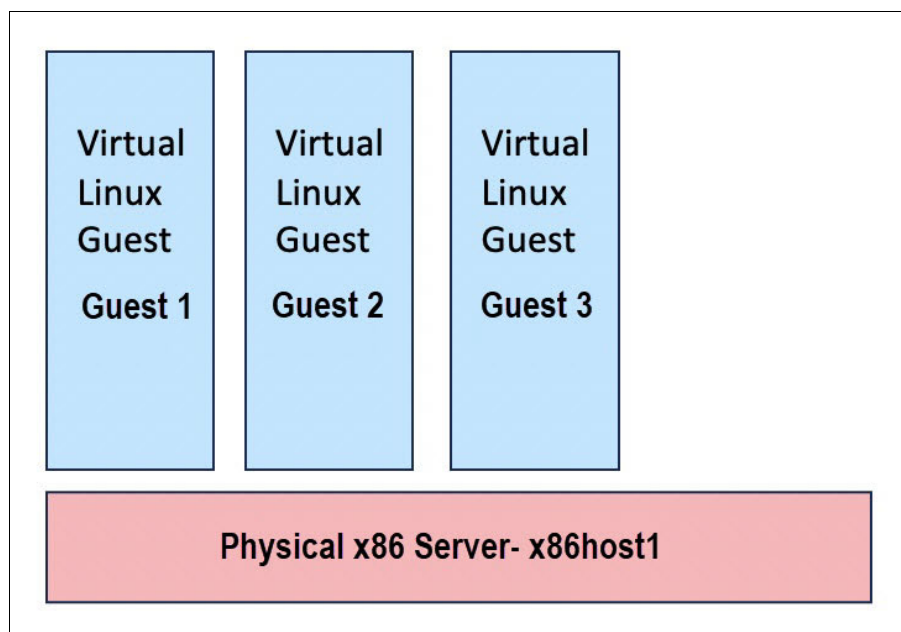


Figure 3-1 Typical x86 virtualization scheme

In a typical x86 deployment of virtual services, the physical servers are generally deployed in pairs or trios, often called clusters of servers. The clusters provide for some standard level of high availability such that if one of the physical servers was to fail, another would be able to take over the running workload with negligible interruption.

3.3 IBM Z virtualization options

Unlike other hardware platforms, IBM Z operate on virtualized hardware by default, which results in incredible performance and efficiency. IBM z/VM and Kernel-based Virtual Machine (KVM) are highly secure and scalable hypervisors to run critical applications and providing the cloud infrastructure.

IBM Z virtualization technology allows you to create virtual processors, communications, memory, I/O, and networking resources. It also simplifies the procedures to provide reliable, highly available, and seamless serviceability for the virtualized infrastructure.

The following are available virtualization options provided by IBM Z:

- ▶ IBM z/VM: IBM virtualization that can be traced back to the beginning of virtualization in computing.
- ▶ KVM: Open-source virtualization that is available for multiple hardware architectures.
- ▶ IBM Processor Resource/Systems Manager (PR/SM) or IBM Dynamic Partition Manager (DPM): Firmware-based virtualization to securely share and partition hardware resources.

- IBM Hyper Protect Virtual Server: A fully encrypted partition with limited and encrypted network access and no access for system administrators.

Figure 3-2 illustrates the scalability offered by IBM Z systems, leveraging advanced virtualization technologies. These technologies enable both horizontal and vertical scalability, allowing organizations to efficiently scale their workload capacity as needed.

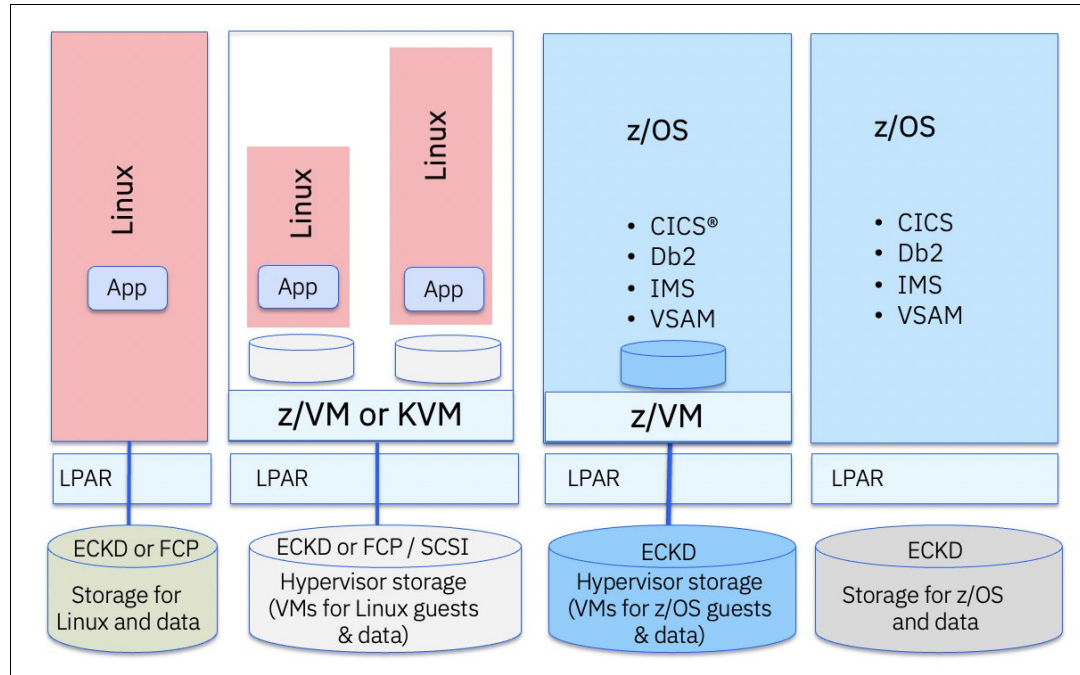


Figure 3-2 IBM Z virtualization options

With virtualization, IBM Z servers can support hundreds to thousands of virtual machines, providing unparalleled efficiency and elasticity. This capability ensures that organizations can dynamically adjust their resource allocation to meet changing demands, optimizing performance and resource utilization.

3.3.1 IBM z/VM

Like virtualization systems deployed that use x86 clusters, IBM Z accomplishes many virtualization functions. Unlike x86, IBM Z does so in a very consolidated and comprehensive way. All of the extensive capabilities of the IBM Z hardware are available for virtualization through the IBM PR/SM hypervisor. But even more extensive capabilities exist when IBM Z virtualization is facilitated by z/VM.

As mentioned in 3.1, “The demand for virtualization” on page 32, virtualization on the IBM mainframe has evolved significantly [since the 1960s](#), culminating in the development of z/VM. Figure 3-3 on page 35 shows the developmental history of z/VM.

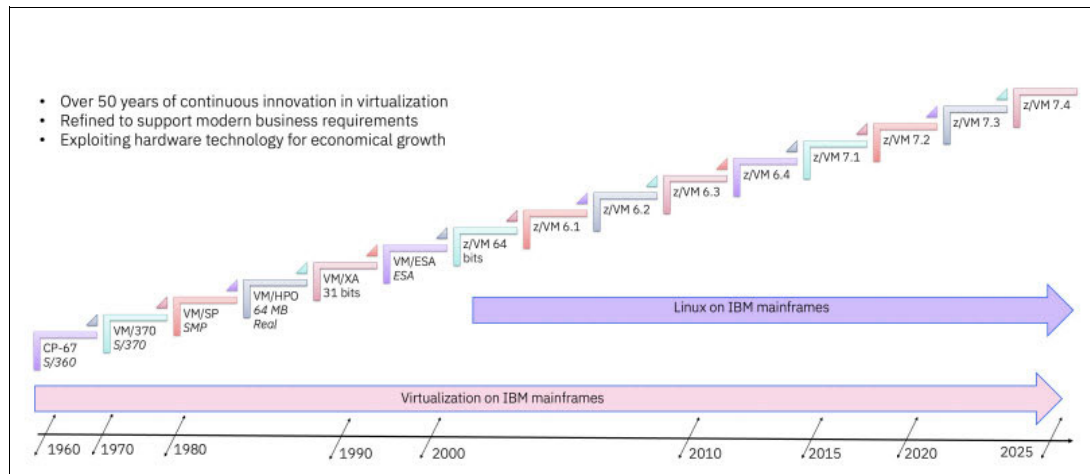


Figure 3-3 Developmental history of z/VM

z/VM is an operating system that facilitates and further enhances the PR/SM hypervisor. A systems administrator may likely know very little about the details of PR/SM. z/VM exposes all of the features and interfaces of the PR/SM hypervisor while further protecting and isolating each virtual machine (VM) from each other and from the physical resources. Make no mistake, Linux can easily operate in a logical partition (LPAR) afforded the general capabilities of PR/SM, and doing so may be an appropriate consideration. However, the virtualization capabilities of z/VM provide added isolation, resource sharing, and resource management features that many systems administrators require.

Figure 3-1 on page 33 provides an example of a typical x86 virtualization system as a model, while Figure 3-4 provides a similar virtualization system as it relates to IBM Z, z/VM, and Linux.

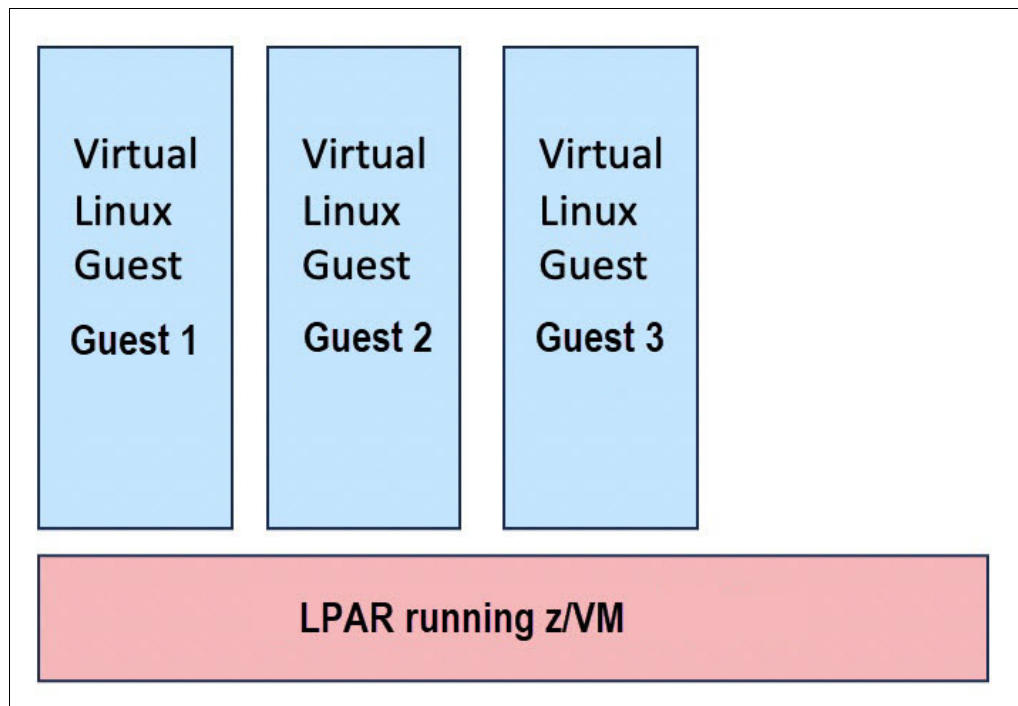


Figure 3-4 Simple z/VM with Linux guests

Running Linux as a guest under z/VM is simple and effectively no different from running z/OS under z/VM. SUSE, Ubuntu and Red Hat all provide Linux distributions that run on IBM Z hardware. The work that IBM has done in collaboration with these major Linux distributions has provided code within the kernel and the core utilities tied to the kernel to facilitate the operation of the Linux kernel with IBM Z hardware. Figure 3-5 illustrates the work that IBM has contributed to the Linux kernel and the Linux operating system to allow Linux to run on IBM Z.

Note: Community distributions such as Alpine, Debian, and Fedora exist in addition to the distributions provided by IBM’s partners. All recent Linux distributions that use GNU Linux kernel version 2.6 or later are technically capable of running on IBM Z. Just keep in mind that the Linux kernel by itself does not make an operating system. To really have a Linux distribution that can run on IBM Z, the distribution must also have binutils, glibc, and other core components built for IBM Z.

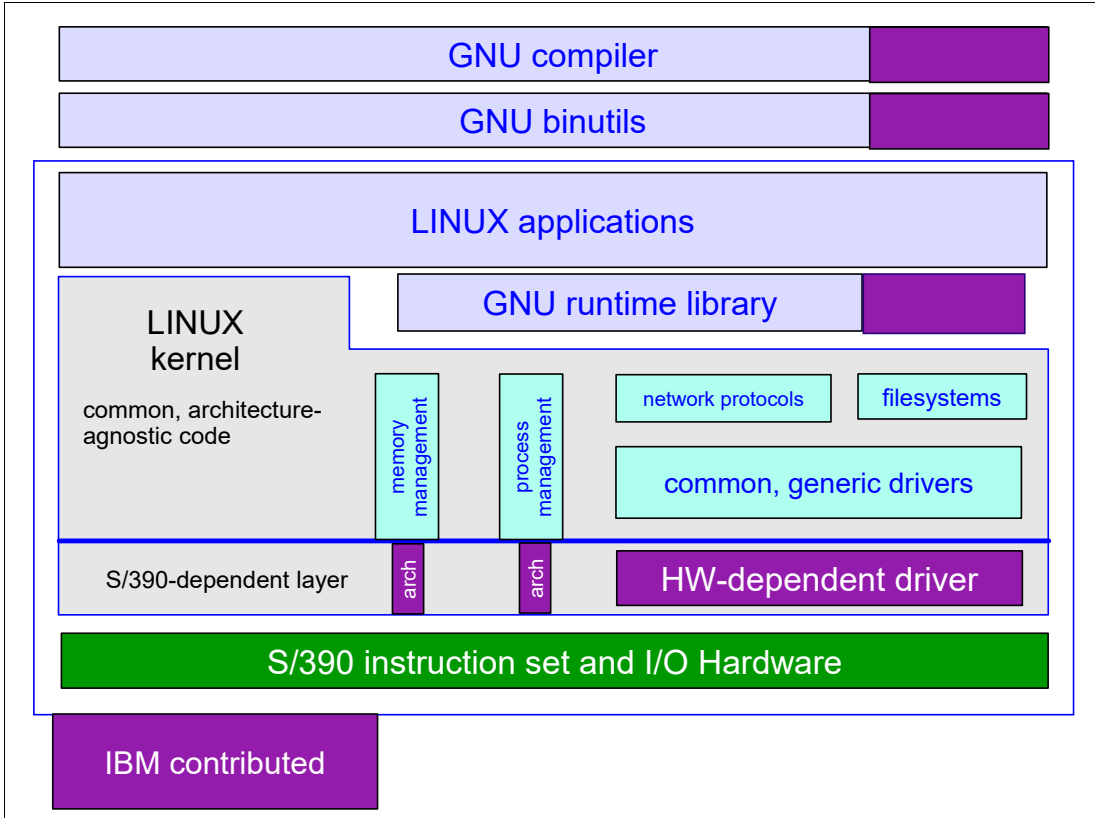


Figure 3-5 Linux kernel and core utilities characteristics on IBM Z

Running Linux on z/VM makes deployment of services faster, often being able to spin up a running Linux server in a matter of minutes. Linux servers can be built, cloned, and deployed within the IBM Z and z/VM infrastructure without the pain of requisitioning, purchasing, mounting, and wiring a new physical server. Development teams who need a new server for a proof of concept can set up and tear down a test environment over and over again with no impact to running production systems. New projects that have completed their development and are ready to be moved into a production environment can do so without the expense of moving or scaling physical resources. Production services can be effortlessly scaled to match the demand, and accommodate all manners of change management.

IBM z/VM serves as the foundation for on-premises cloud computing on IBM Z, offering a robust virtualization platform with numerous benefits:

- ▶ z/VM allows organizations to host Red Hat OpenShift in virtual machines, providing a flexible and scalable environment for containerized workloads.
- ▶ With z/VM, users can host various operating systems with diverse workloads in virtual servers, enabling efficient resource utilization and workload consolidation.
- ▶ z/VM virtualize and shares resources with exceptionally high levels of utilization, maximizing the efficiency of the underlying hardware infrastructure.
- ▶ Organizations can quickly benefit from new features and enhancements through the continuous delivery model of z/VM, ensuring they stay up to date with the latest advancements in virtualization technology.
- ▶ Paired with IBM Cloud Infrastructure Center, z/VM forms the foundation for on-premises cloud solutions on IBM Z, offering extreme scalability, security, and efficiency for cloud-based workloads.
- ▶ z/VM provides support for IBM z16 architecture, including features like Crypto Express8S for guests and IBM System Recover Boost for improved processor performance during workload operations.
- ▶ z/VM delivers a release cadence that enhances its continuous delivery model, with features like Single System Image (SSI) support in the base product, improving the availability of z/VM systems and minimizing disruptions to critical applications.

For more information, [The Virtualization Cookbook for IBM Z Volume 1: IBM z/VM 7.2, SG24-8147](#).

3.3.2 KVM on IBM Z

KVM holds strategic importance as an open-source component for the IBM Z platforms.

IBM actively contributes to the KVM enablement for IBM Z within the open-source community, facilitating distribution partners to deliver KVM on these platforms. Notably, several Linux distributions offer support for KVM on IBM Z, including [Red Hat Enterprise Linux](#), [Ubuntu](#), and [SUSE Linux Enterprise Server](#). These distributions ensure compatibility and provide support for KVM deployments on IBM Z, enabling organizations to leverage the benefits of virtualization technology on these powerful platforms.

KVM on IBM Z offers a powerful virtualization solution with a range of features and benefits:

- ▶ KVM enables the pass-through of Crypto Express adapter domains in KVM guests, allowing for secure cryptographic operations within virtualized environments.
- ▶ KVM supports on-chip compression for "pervasive usage" with Linux guests, enhancing data storage efficiency and performance within VMs.
- ▶ With support for up to 8TB (and 16TB) of host memory, KVM on IBM Z enables organizations to run memory-intensive workloads with ease, ensuring optimal performance and scalability.
- ▶ KVM leverages new vector instructions to enhance performance across various workloads, providing faster execution and improved resource utilization.
- ▶ KVM ensures the security and protection of business data by leveraging elliptic-curve cryptography (ECC), a robust encryption technique that safeguards sensitive information within virtualized environments.
- ▶ KVM enables IBM Secure Execution for Linux to protect and encapsulate confidential workloads from access and modification of the Hypervisor administrator.

- ▶ As an integral part of the Linux operating system, KVM inherits all Linux exploitation capabilities, serving as a reliable and seamless host for virtual machines.

For more information on these distributions, see:

- ▶ [The Virtualization Cookbook for IBM Z Volume 2: Red Hat Enterprise Linux 8.2, SG24-8303](#)
- ▶ [The Virtualization Cookbook for IBM z Systems Volume 3: SUSE Linux Enterprise Server 12, SG24-8890](#)
- ▶ [The Virtualization Cookbook for IBM z Systems Volume 4: Ubuntu Server 16.04, SG24-8354](#)

3.3.3 PR/SM and DPM

PR/SM (Processor Resource/Systems Manager) and DPM (Dynamic Partitioning Manager) are key features of IBM Z, offering dynamic resource management and provisioning capabilities.

Note: PR/SM is the same hypervisor that underpins both the traditional mode of operation and DPM. DPM is a mode of operation of the HMC and firmware in the way that configuration of PR/SM is achieved. Because DPM represents such a different view of management of an IBM Z server it is usually referred to as a different mode of operation (as we have done in this book).

The important thing to remember is that a machine in DPM mode and a machine in traditional mode will treat workloads the same, since PR/SM is used in both cases.

DPM simplifies the provisioning and management of resources on IBM Z systems, enables administrators to dynamically allocate and deallocate resources such as processor cores, memory, and I/O adapters to logical partitions (LPARs) based on workload demands and helps optimize resource utilization and improve system efficiency by dynamically adjusting resource allocations to meet changing workload requirements.

PR/SM manages and virtualizes all the installed and enabled system resources as a single large SMP (Symmetric Multiprocessing) system and enables full sharing and partitioning of installed resources among multiple logical partitions (LPARs) with high efficiency and utilization levels.

- ▶ PR/SM and IBM DPM offer robust virtualization capabilities on IBM Z, providing organizations with the flexibility and efficiency needed to manage their workloads effectively:
- ▶ PR/SM manages and virtualizes all installed and enabled system resources, treating them as a single large SMP system. This allows for full sharing and partitioning of resources, ensuring the highest levels of efficiency and utilization.
- ▶ You can scale up or scale out on demand, with support for up to 85 LPARs. This scalability enables businesses to adapt to changing workload demands without compromising performance or resource availability.
- ▶ IBM DPM simplifies the provisioning and management experience, allowing administrators to efficiently allocate resources to different partitions based on workload requirements. This dynamic management ensures optimal resource utilization and responsiveness.
- ▶ PR/SM and IBM DPM provide workload isolation, with a design tailored for the highest security certification (EAL5+). This ensures that workloads remain secure and isolated from each other, protecting sensitive data and applications.

The virtualization capabilities of IBM Z continue to evolve with new dynamic optimization and scalability enhancements. These improvements further enhance performance, scalability, and resource efficiency, enabling organizations to meet the demands of modern workloads effectively.

3.4 Single system image and live guest relocation

A critical responsibility of systems administrators is ensuring that all systems are running the latest operating systems software and that all maintenance and security fixes have been applied. Protecting the system from unexpected downtime and from security vulnerabilities, ensuring that applications are running at the latest patch release levels, and balancing loads across a diverse infrastructure are all tasks that keep systems administrators awake at night. This is a particularly troubling challenge in the data center, where downtime must be minimized and maintenance windows are scarce.

General virtualization does not accommodate the ability to take down the host system to apply maintenance updates. If you have a critical problem on the host system, all the guests running on the host will likewise need to be taken down in order to reboot the host. This is clearly not an ideal scenario.

The z/VM single system image (SSI) is a clustering technology that provides multiple, redundant host systems upon which virtualized guests run. Each member of the SSI cluster shares a common pool of DASD volumes, minidisks, network devices, and user data. Ideally the cluster members would be contained on separate CECs for optimum safety if a failure were to occur, although running the members on the same CEC is also feasible. The members of the SSI cluster are managed together.

Coupled with SSI is live guest relocation (LGR), which facilitates the relocation of a Linux guest from one member of the SSI cluster to another. This relocation happens nearly instantaneously, without the Linux guest having any knowledge of the relocation. Network processes and connections, disk operations, and user interactions on the Linux guest are completely unaware that the underlying infrastructure has moved to a different “physical” environment. Figure 3-6 depicts a very simple representation of an SSI cluster composed of two members, zhost73a and zhost73b. zhost73a is currently hosting three Linux guests while zhost73b hosts a single Linux guest.

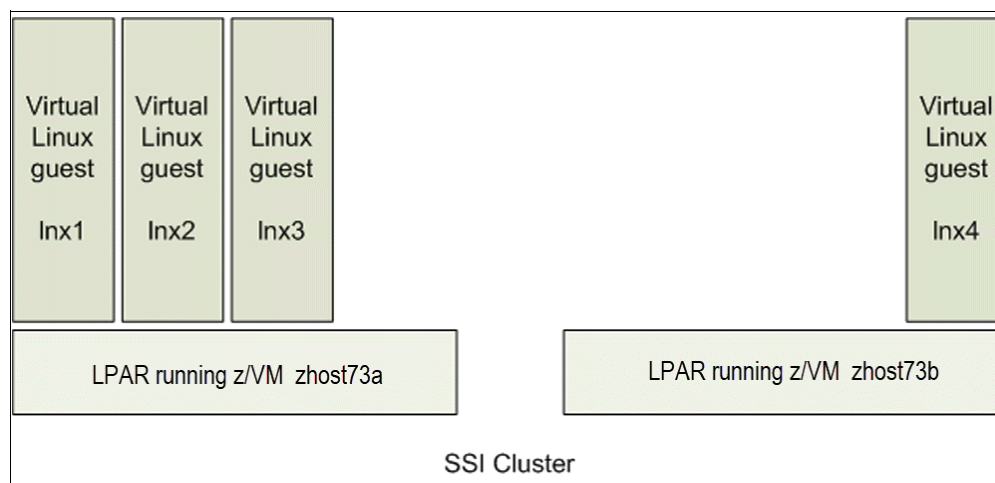


Figure 3-6 Simple representation of SSI cluster before live guest relocation

The relocation of Linux guests from one SSI member to another makes it possible to perform maintenance on the individual SSI cluster members without disrupting the services running on the Linux guests. With all Linux guests relocated away from an SSI member, that SSI member can now be updated and rebooted, with no impact at all to any running guests. When the maintenance on this SSI member is completed, Linux guests can be relocated back to their original host members. Perhaps all Linux guest systems could be relocated to this SSI member while similar maintenance is performed on other SSI members in the cluster.

An additional benefit of SSI and LGR is the ability to relocate workloads to accommodate a more balanced use of system resources. If an SSI cluster currently contains a configuration of multiple Linux guests that are overusing the network, a portion of the guests could be relocated to a different member of the SSI cluster where network utilization is lower.

Figure 3-7 shows that a Linux guest has been relocated from zhost73b to zhost73a with no interruption in the services that are running from the Linux guest. Now that there are no guests running on zhost73b, the host can be rebooted. After rebooting zhost73b, Linux guests can be relocated back onto zhost73b.

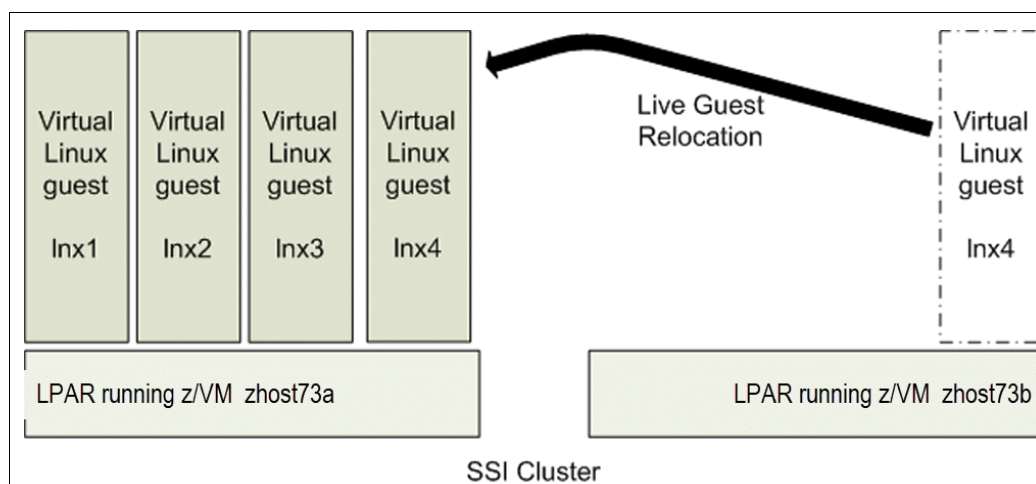


Figure 3-7 A simple representation of live guest relocation of a Linux guest

This convenient mechanism of relocating guests with LGR among the various SSI cluster members is precisely the flexibility that the systems administrator needs in order to keep systems up to date while minimizing downtime, while also giving the administrator the ability to move workloads more freely within the infrastructure to make the best use of resources.

More to the point, knowing that z/VM, SSI, and LGR can be used in this way makes the decision to migrate workloads to Linux on IBM Z all the more compelling.

This section provides merely a brief overview of SSI and LGR. There are several IBM Redbooks publications that describe SSI and LGR in greater detail. Such titles include (but are not limited to):

- ▶ [An Introduction to z/VM Single System Image \(SSI\) and Live Guest Relocation \(LGR\), SG24-8006](#)
- ▶ [The Virtualization Cookbook for IBM Z Volume 1: IBM z/VM 7.2, SG24-8147](#)
- ▶ [Using z/VM v 6.2 Single System Image \(SSI\) and Live Guest Relocation \(LGR\), SG24-8039](#)

3.5 z/VM hypervisor components

Two primary components of z/VM help PR/SM in managing the virtualization environments:

- ▶ The z/VM Control Program (CP)
- ▶ The z/VM Conversational Monitor System (CMS)

These components are command line operating environments that give the system administrator control over the hypervisor. An additional component, the IBM Cloud Infrastructure Center (ICIC), allows a graphical interface that can be used for easier administration over the hypervisor and its guests. ICIC can help minimize terminal and command interactions for daily guest management chores; however, z/VM knowledge is still required from an effective infrastructure architecture management perspective, as well as for initial set up tasks.

3.5.1 Control program

The control program (CP) provides a guest (in this example, the Linux operating system, although z/VM virtualization supports other operating systems) with a complete VM environment with virtual resources that appear as real hardware resources.

Communication with the control program is through CP commands that are used by the z/VM administrator and Linux administrator to manage, query, and allow the definition of additional resources.

When a Linux guest logs on to a z/VM session, it starts its own CP session. For production systems, this login is usually done automatically when the z/VM system is initially loaded or booted. An entry exists in the z/VM directory for each VM that can be started.

Each entry contains detailed information about each user or guest and the virtual resources that are required by the guest operating system, as well as details of the relative priority of the virtual machine. This information is used by CP to determine which VM is to be dispatched. Communication to the Linux system can be through the Linux VM console (which must be a 3270-type terminal emulator), or more commonly by using an SSH client terminal.

Note: If an administrator logs off the Linux VM console by using the conventional **LOGOFF** CP command, the guest machine ends the session at the z/VM host level, causing the virtual machine to power off and terminate all running work. The administrator must use the **DISCONNECT** command (not the **LOGOFF** command) to only remove the connection from the terminal to the console prompt and ensure the guest remains running in the background. This slight difference between **LOGOFF** and **DISCONNECT** can stop a running system.

3.5.2 Conversational Monitor System

The Conversational Monitor System (CMS) is a single user operating system that runs only as a z/VM guest. CMS is used by the z/VM system administrator to manage the system components and to create and edit VM user profile entries in the z/VM environment. CMS provides an interactive environment for z/VM administration. Many instances of the CMS operating system support service machines, such as TCP/IP, print services, directory maintenance, accounting, and error recording.

For more information about z/VM, refer to [Introduction to the New Mainframe: z/VM Basics](#), SG24-7316.

CP and CMS give the system administrator a more direct route to manipulating the available resources for the benefit of the Linux guest

3.5.3 Log-On Wave for IBM Z

[Wave for IBM Z](#) is an intuitive virtualization management software product that provides management, administration, provisioning, and enables automation of Linux virtual servers in a z/VM environment. Wave for IBM Z simplifies and accelerates the management and daily administration of highly virtualized z/VM and Linux server environments on IBM Z.

To reduce the complexity of z/VM management, Wave for IBM Z provides a solution to help system administrators in their daily tasks. The following is a list of features that can help with maintenance tasks:

- ▶ Display and manage virtual servers and resources, all from the convenience of a single graphical interface
- ▶ Provision VMs, and install a guest operating system
- ▶ Provision virtual resources, such as processors, memory, network, and storage
- ▶ Capture and clone virtual servers across partitions
- ▶ Create and configure virtual switches (vSwitches) and guest LANs
- ▶ Relocate VMs to other partitions
- ▶ Display, monitor, and manage z/VM hypervisor resources, such as paging

3.5.4 IBM Cloud Infrastructure Center

IBM Cloud Infrastructure Center has been presented earlier in the chapter, since it also provides a management solution for KVM hypervisor guests. Furthermore, guests belonging to KVM and z/VM can be managed together from the same tool.

IBM Cloud Infrastructure Center is an advanced infrastructure management offering that provides on-premises cloud deployments of z/VM and Red Hat Linux KVM based Linux VMs (Red Hat, Ubuntu or SUSE) on the IBM Z platform and the integration to higher-level cloud automation tools, such as IBM Cloud Automation Manager or VMware vRealize.

It constitutes an alternative for former IBM Wave users for z/VM guest management that employed the tool to reduce the complexity of their daily system administration tasks. This tool would be covering those same needs providing further options.

IBM Cloud Infrastructure Center includes the following features:

- ▶ Simplification of VMs lifecycle infrastructure-as-a-service management tasks.
- ▶ Availability of a self-service portal for easy workload deployment.
- ▶ Provisioning of VMs, and installation of guest operating systems.
- ▶ Enablement of software-defined infrastructure.
- ▶ Integration of cloud management and Red Hat OpenShift by way of OpenStack compatible APIs.
- ▶ Display and management of virtual servers and resources, all from the convenience of a single interface.
- ▶ Capturing and cloning of virtual servers across partitions.

- ▶ Provisioning of VMs virtual resources, network, storage and compute, such as processors and memory.
- ▶ Creation and configuration of virtual switches (vSwitches) and guest LANs of various types.
- ▶ Relocation of VMs to other partitions.
- ▶ Creation of storage volumes, including boot volumes, data volumes and consistency groups backups.

For more information, see [IBM Cloud Infrastructure Center](#) and [IBM Cloud Infrastructure Center documentation](#).

3.6 Virtualized resources

A key feature of IBM Z is how resource utilization is optimized and maximized. In the current environment of distributed computing, the RAM, the CPU, or the disk is underutilized most of the time that the server is running, but is necessary to have available when the server is under peak load. With IBM Z, a considerable amount of “overcommitting” is possible, such that RAM, CPU, and I/O can adequately accommodate the workload when the workload needs it, and the resources can be diverted elsewhere, without having to commit specific resources to any one workload. Although resources can be rigidly committed to a specific workload, it is the flexibility of the virtual resources that is so appealing. Overcommitting is quite powerful for z/VM virtualization because typically not every guest will need all of its allocated resources all at the same time.

3.6.1 Virtualized CPU

Whether running Linux directly in an LPAR or in z/VM, the guest needs processing power. IBM Z offers a few choices of processors. When Linux is the chosen operating system for the workloads, system administrators generally choose to employ an Integrated Facility for Linux (IFL) Processing Unit (PU) due to its attractive price point. Where a mix of Linux and other systems coexist, a standard central processor (CP) is the more appropriate choice. Other specialty CPs exist, such as the zAAP for running dedicated Java workloads.

“z/VM Mode LPAR” allows different processor types to be mixed in the same LPAR. This provides greater flexibility in supporting a variety of workloads in the same LPAR. Instead of running z/OS and Linux in separate z/VM systems, for example, a single z/VM could host all the guests. This provides greater efficiencies in the sharing of resources, and when workloads in the various guests require specific affinities (such as a Red Hat OpenShift cluster using database services on z/OS Db2), some performance advantages can be realized.

The number of IFLs or CPs on the machine reflect directly on the performance of the Linux guest running in an LPAR. The number of virtual CPUs allocated to a single Linux guest should not exceed the number of logical CPUs allocated to the LPAR. For example, if the LPAR has four IFLs, then do not allocate five virtual CPUs to a single Linux guest machine. If a situation occurs where the Linux guest uses 100% of the CPUs, that will adversely affect the entire LPAR. However, in an LPAR with four IFLs, you can assign three virtual CPUs to a LinuxA guest and two virtual CPUs to a LinuxB guest, as well as another two virtual CPUs to a LinuxC guest.

All requests for CPU cycles will be managed by z/VM according to the relative priorities of the Linux guests. CPU configuration best practice is to maintain the ratio of four active virtual CPUs to one logical CPU allocated to the LPAR.

3.6.2 Virtualized disk

IBM Z disk storage is commonly referred to as a direct access storage device (DASD). Mainframe system administrators have a long and unique history with DASD. Traditionally, IBM Z has supported only IBM extended count key data (ECKD) DASD, which was a developed from Count Key Data (CKD) devices to provide improved performance for Fibre Channel-connected DASD.

The ECKD devices are defined as one of three 3390 DASD models, each of different sizes. The models, and their capacity as measured in cylinders and in megabytes, are listed in Table 3-1.

Table 3-1 Some standard 3390 DASD models

Model	Cylinders	Storage Capacity
Model-3	3,339	2.83 GB
Model-9	10,017	8.51 GB
Model-27	32,760	27.84 GB
Model-54	65,520	55.68 GB

In addition to the sizes of 3390-27 and 3390-54 shown above, some DASD devices are defined with sizes that follow the original multiples of the 3390 Model 1 (which had 1,113 cylinders). You may see Model-27s with 30,051 cylinders, or Model-54s with 60,102 cylinders. Other than them being slightly smaller and having less usable capacity than the “Ki”-based sizes above, they are perfectly valid.

Modern ECKD controllers also support the Extended Address Volume (EAV). An EAV is known as a 3390 Model A, and can be defined to be larger than 65,520 cylinders. They can have an arbitrary number of cylinders, up to the current architectural limit that yields about 1 TB of space.

Although DASD is common in the mainframe world, it is not well known in the distributed x86 world.

Many x86 computing environments will have disk storage maintained in Storage Attached Networks (SANs) and other similar, external storage arrays. IBM Z is fully capable of using disk storage from a SAN or network-attached storage (NAS). In many cases, the system administrator chooses to maintain the data of a particular application on the storage array while choosing to migrate the application workload to IBM Z. Whether maintaining the data on a SAN or migrating the data to IBM Z storage, the virtualized disk can be readily accessed by the workloads in the virtual environment. In many cases, leaving the data intact on the SAN will ease and simplify the migration effort.

With z/VM and Linux on IBM Z, disk device support is expanded to fixed-block architecture (FBA) DASD and also to Small Computer System Interface (SCSI). FBA and SCSI disks are connected to IBM Z via the Fibre Channel Protocol (FCP). The connection to SCSI devices is managed by the zFCP Linux module driver. The SCSI devices are usually dedicated to the Linux guests.

It is good practice to use ECKD DASD to boot Linux and for static data, and use SCSI FCP for data applications. DASD is well suited to booting and hosting the operating system, but systems administrators may find that performance from the SCSI FCP disks is better adapted for data.

Note: With the High Performance IBM FICON® (HPF) feature, ECKD DASD performs quite comparably to most FCP-attached disks. When considered from a price/performance perspective however, FCP can be more attractive. Any additional cost of ECKD can be made up when extremely high availability is required, through the use of IBM GDPS® to manage replication and failover.

Disk storage, by itself, is not really a virtual resource. The bits and stripes on the disk do not have the same characteristics for virtualization that memory does. Disk is a more permanent resource than memory. Nevertheless, allocating free disk space for a workload should be just as flexible and effortless as allocating virtual processing power or virtual memory. A competent hypervisor facilitates the management of disk storage.

For a more detailed description about disk storage, see 6.2, “Storage analysis” on page 84. For more information about z/VM and disks, see [Introduction to the New Mainframe: z/VM Basics, SG24-7316](#).

3.6.3 Virtualized memory

System memory (to use the Linux term) or storage (to use the z/VM term) is a resource that is shared across all z/VM guests. Each virtual guest is assigned a defined amount of virtual storage during login.

The key to efficient memory management is to be aware of the total amount of virtual memory that is likely to be active at any time, and also be aware of the amount of real memory (storage) that is allocated to the z/VM LPAR.

z/VM allows you to overcommit memory, but keep the overcommitment ratio of the total amount of virtual memory likely to be active to the total amount of virtual memory to around 2:1. For test or development workloads, the ratio should be no more than 3:1.

The keys to determining the appropriate virtual memory size are to understand the working set for each virtual machine, and to ensure that the Linux images do not have any unneeded processes installed. Another recommendation is to use VDisks for swap, as described in “Swap device consideration” on page 47.

Memory management features

There are memory management features for Linux and z/VM that you can use to reduce the amount of memory required by virtual guests:

- ▶ Cooperative Memory Management (CMM)
- ▶ Collaborative Memory Management Assist (CMMA)
- ▶ Named Saved System (NSS)
- ▶ Discontiguous Saved Segment (DCSS)

These are features that simply are not possible in a distributed x86 environment. Only z/VM can provide these versatile features, dramatically reducing the amount of physical memory required to maintain a similar set of workloads.

CMM

CMM is used to reduce double paging that may happen between Linux and CP. CMM requires the IBM Virtual Machine Resource Manager (VMRM) running on z/VM to collect performance data and notify the Linux guest about the constraints when they occur. On Linux servers the `cmm` kernel extension is required, and it is loaded with the `modprobe` command.

CMMA

CMMA enables CP and Linux to share the page status of all 4 KB pages of guest memory. Linux does this by marking the status of each page; this allows CP to preferentially steal unused and volatile pages and thus reduce paging.

NSS

NSS allows virtual guests to share a read-only copy of a single operating system such as CMS or Linux. The benefit of this feature is that only one copy of the operating system resides in storage accessible by all virtual machines. This decreases storage requirements and simplifies maintenance.

DCSS

DCSS allows virtual machines to share reentrant code for applications, such as Oracle, which also reduces overall storage requirements. Figure 3-8 illustrates how both NSS and DCSS work. There is one copy of the application in real storage and Linux guests use this single copy. The NSS copy of Linux is also shared by all virtual guests.

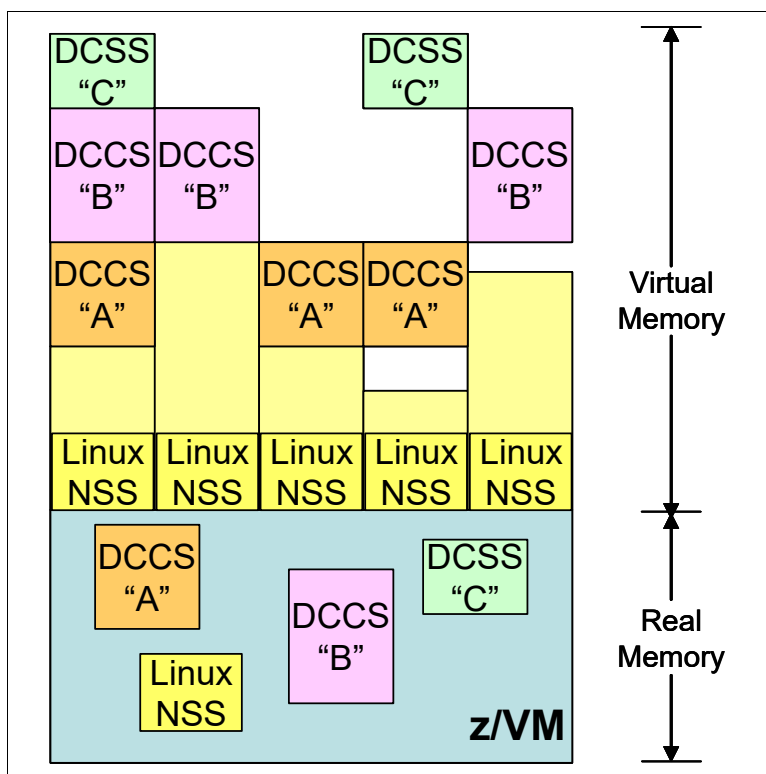


Figure 3-8 DCSS and NSS shared by multiple Linux virtual guests

For more information about setting up a Discontiguous Saved Segment and using the Execute-In-Place (XIP) file system, refer to [Using Discontiguous Shared Segments and XIP2 Filesystems With Oracle Database 10g on Linux for IBM System z](#), SG24-7285.

Note: When defining memory requirements for virtual Linux guests, remember that the Linux kernel will use all the extra available memory allocated to it as a file system cache. Although this is useful on a stand-alone system (where that memory would otherwise go unused), in a shared resource environment such as z/VM this causes the memory resource to be consumed in the LPAR. Therefore, it is important to assign only the memory needed for the running applications when they are at peak load.

Linux swap should be thought of as an overflow when an application cannot get enough memory resource. Thus, when paging occurs, this is an indication that either more memory needs to be assigned or the application needs to be analyzed to understand why more memory is needed.

Swap device consideration

Understand that the concept of “swapping” is different today than when it was invented, back when large amounts of RAM were ridiculously expensive. Modern operating system memory technology is more focused on paging than swapping. As suggested in the note a few paragraphs back, it is a best practice to commit a specific amount of virtual memory to each Linux guest to accommodate no more than its intended workload, and to fine tailor this amount of memory precisely so that paging does not normally occur. This may not be realistic, but it is a principle to seriously consider.

In the absence of the perfect memory configuration, and when workloads demand significant swapping, the ideal is to provide a VDisk device for this purpose. VDisks are virtual disks allocated in memory, and they become a fast swap device for Linux. Swapping to a VDisk in memory is far more efficient than swapping to DASD, and it is generally less expensive, too, considering all factors. The Linux administrator must take care during the initial installation of the Linux guest to ensure that the VDisk is formatted as a swap device. But more than that, the VDisk must also be formatted each time the Linux guest is booted.

For more information about optimizing memory on z/VM and Linux, see [Linux on IBM System z: Performance Measurement and Tuning](#), SG24-6926.

3.6.4 Virtualized network

The physical network in IBM Z consists of devices known as Open Systems Adapters (OSAs). IBM z16 leverages the Open Systems Adapter-Express (OSA-Express7S and [OSA-Express6S](#)) and RoCE Express features. With up to 96 OSA-Express7S 1000BASE-T ports and up to 48 OSA-Express7S 25 GbE SR ports, the z16 ensures robust connectivity to external networks.

As might be expected, the z/VM feature to access the Internet Protocol network is TCP/IP for z/VM. OSA-Express devices can be virtualized through a virtual switch (vSwitch) device to many Linux guests. It is available using special z/VM machines known as vSwitch controllers. Each Linux guest connects using a virtual device controlled by the qeth module to a virtual switch system in a z/VM LPAR.

An important benefit of the vSwitch system is that it can be set up with redundant OSA devices that provide a failover network system on z/VM.

HiperSockets provide high-speed interconnectivity among guests running on a IBM Z. This technology does not require any special physical device configurations or cabling. The guests simply communicate with one another internally via the in-memory capabilities of the PR/SM hypervisor. HiperSockets, however, are not intended to be used for sophisticated networking and should not be used for external traffic.

Both OSA-Express and HiperSockets use the Queue Direct I/O (QDIO) mechanism to transfer data. This mechanism improves the response time using system memory queues to manage the data queue and transfer between z/VM and the network device. Various examples are available in 6.1, “Network analysis” on page 72.

More recent IBM Z servers have introduced RDMA over Converged Ethernet (RoCE) through the ROCE-Express adapters. When used by Linux on IBM Z, a RoCE Adapter can be used for either standard TCP/IP communication or for RoCE. RoCE Express is also referred to as Shared Memory Communications (SMC). In addition to RoCE (which is also called Shared Memory Communications-RoCE, or SMC-R), there is a firmware-internal version called Shared Memory Communications-Direct, or SMC-D. SMC-R and SMC-D will offer high performance communication methods for Linux workloads.

For more information about network in Linux and z/VM, see [Advanced Networking Concepts Applied Using Linux on IBM System z](#), SG24-7995.



Migration process

In the field of information technology, migration refers to the process of moving from one operating environment to another. In many cases, the move to a new platform involves various organizational and strategic changes.

This chapter provides you with information regarding the approaches involved in planning your migration and defines various types of stakeholders along with their roles and responsibilities. Not every organization uses the same titles for stakeholders as those listed here, but the titles that you use should match the functions described.

Additionally, this chapter describes the process that should be used when undergoing a migration project from identifying the stakeholders, assembling them, and identifying success criteria through to verifying both the migration itself and its success.

The following main sections are available in this chapter:

- ▶ 4.1, “Stakeholder definitions” on page 50
- ▶ 4.2, “Identify the stakeholders” on page 53
- ▶ 4.3, “Assembling the stakeholders” on page 54
- ▶ 4.4, “Migration methodology” on page 55

4.1 Stakeholder definitions

This section categorizes stakeholders as comparatively non-technical business stakeholders, or as more technically oriented information technology stakeholders. A stakeholder is anyone who is affected by the activities of the project. Conversely, it could also be stated that a stakeholder is anyone who affects the migration project. A stakeholder analysis is essential to facilitate the communication and cooperation between the project participants and to assure successful outcomes, whether the outcomes are individual milestones or the entire completed project. Ensure that stakeholders are involved during the planning stages of the migration project, rather than simply when they are needed to perform tasks for you in the execution stages of project migration.

4.1.1 Business stakeholders

Business stakeholders are those who are responsible for making the decisions about the business and provide direction for migration:

- ▶ **Business owners or business managers**

These stakeholders lead business lines such as Chief Financial Officer (CFO), marketing, and sales. They are concerned with the business and financial resources used in the project. They often view information technology as a tool to accomplish business tasks efficiently and effectively. These stakeholders may have a staff member reporting on technical issues, including migration proposals, that must be evaluated by the technology stakeholders. Conversely, proposals for migration may originate with the technology stakeholders, who must provide sufficient justification to the business owner. Migration justifications are discussed in Chapter 2, “Analyze and understand” on page 21.

Large and complex consolidation projects require participation from several business owners and business lines. The business owners and IT management must be closely aligned and cooperate openly to achieve a successful migration.

- ▶ **Business managers and supervisors**

These stakeholders are concerned with the workflow within their departments. They understand the importance of the application and how their employees use it. They select users who are the most qualified and motivated to participate in the migration project.

- ▶ **Quality Auditors**

Large and complex consolidation projects require participation from quality auditors to create the Quality Indicators (QI) and ensure that the QI is achieved.

- ▶ **Users**

These stakeholders are the end customers. They use the application or consume the services provided by the application and perform testing to assure that the application is working at least as the same level after the successful implementation of the migrated system. In a migration without enhancements, users should not see any changes. Availability and response times should meet the service level objectives agreed to by management and communicated to the users. Their perspective and input to the conversion project is valuable. Their satisfaction must be a criteria for the success of the migration project.

4.1.2 Operational stakeholders

Operational stakeholders are different from Business stakeholders in that these are the people who are responsible for implementing the systems and changes:

- Chief Information Officer (CIO)

The highest level of IT management is usually the Chief Information Officer (CIO). In some companies, the highest level of IT management may be a director or a manager. This stakeholder's role is to provide vision and leadership for information technology initiatives. The main concerns are to support business operations and services as well as to improve cost effectiveness, improve service quality, and develop new business process services. These stakeholders should clearly understand the benefits and risks of the migration project.

- Project manager (PM)

This stakeholder has the responsibility of creating and managing the plans, interdependencies, schedule, budget, and required personnel for the migration effort.

Other responsibilities include defining and obtaining agreement on the approach. The project manager tracks and reports to all key stakeholders on progress against plans, escalating any issues or risks where appropriate.

- IT managers and supervisors

Some stakeholders will be managers or supervisors of mainframe system administrators and system programmers. Managers at this level will have various types of influence on the migration project. Some projects may be originated and championed by these stakeholders. They usually have a high level of technical competence and understanding of the technologies that will be used in the migration project. These stakeholders should be intimately aware of the staffing and training considerations of the migration project. They should work closely with their staff to assess current skills and map out a training plan to acquire the required hardware and software-related skills.

- Mainframe system administrator, system programmer

The mainframe system administrator is responsible for setting up hardware definitions. The hardware components defined are CHPIDs (channels), control units, and devices. A channel is a generic term for external I/O communication paths to Open Systems Adapter (OSA) for Ethernet networks, IBM FICON or Fibre Channel Protocol (FCP) for attached disk, printers, tapes, and consoles. System programmers install and maintain z/VM including defining LPARs, user directories, and resources for CMS users and Linux guests. They also configure the network connections, virtual switches, and installation of additional products and services such as the IBM Performance Toolkit for VM.

To maintain and execute these tasks, they have an option to use IBM Wave for z/VM that simplifies the management of z/VM and Linux guests providing an intuitive graphical interface.

- UNIX, Linux, and Windows system administrators

Linux administrators may assist in installing Linux on IBM Z, or take over administration tasks after the Linux guest has been installed. These stakeholders work closely with the system programmers when major configuration changes or additions are made (such as increasing the memory, disk space, or CPU). All other Linux administration duties will be the same as on other platforms, such as Linux on x86.

Various other Windows and UNIX administrators will be involved in the migration project. This is partially dependent upon where the source system is hosted (that is, the platform where the source application resides). The administrator of the source system will be heavily involved because that is the application that is being migrated.

Other services, such as DNS, mail servers, and security, will run on UNIX or MS Windows servers. These and other services will usually be required by the application that is being migrated. The administrators of these services will be required to make adjustments for the migrated application.

- Network engineers

These stakeholders design, install, and maintain data communication equipment, such as routers, switches, local area networks (LANs), wide area networks (WANs), and other network appliances. They monitor the network for performance and errors. During migration, network engineers help to design the new network and deploy any changes to the existing network.

Network engineers must be familiar with the communications components that are unique to Linux on IBM Z like vSwitch, for example. For more information about IBM Z networking, refer to 6.1, “Network analysis” on page 72. The network concepts and tools outside of the IBM Z box is the same for these stakeholders.

- Database administrators (DBA)

The tasks performed by these stakeholders can be separated into two or more different but related job functions such as database analyst, database administrator, and system administrator. The Database administrators are responsible for installing and maintaining the database management system (DBMS) code base. They design and implement the corporate databases, assure the data integrity, and good database performance. They work closely with the application development group to ensure that the application is running efficiently.

- Application architects and developers

Applications developed in-house require porting and testing on the target Linux system. The effort involved can vary greatly, depending on what language the application is written in and how hardware-dependent the code is. Open source and commercial tools are available to help with tasks such as assessing the portability of your applications. IBM Global Services, as part of its migration services offerings, uses tools developed in cooperation with IBM Research® to help with code assessment and conversion. The application architect and developers are the stakeholders who are responsible for this porting effort. Refer to 6.3, “VMware to KVM migration options” on page 94 for more information about the issues that need to be considered.

- Operators

The operators monitor the application, the operating, and the physical environment by checking the monitor consoles, logs, alerts. They raise problem tickets, notify support teams, and escalate issues to management. New tools and procedures that result from the migration project are required to them.

- Service Desk staff

These stakeholders are on the front line of support to the customer. They are usually the first ones to get a call when there is a real or perceived problem with the application. They need to be the first staff trained on the new environment, and should be heavily involved in the migration testing so they can provide meaningful support after the migration.

- Users

Perhaps the most important stakeholders involved in a migration are those who will use the application every day. They need to be involved from the beginning because the success of the project will depend in large measure on how easy the system is for them to use. Ideally, it should have the same “look and feel” to which they are accustomed. However, in many cases a migration is often an opportunity for firms to improve the application, which often results in additional functions and procedures that they need to learn.

Note: Users are identified both as Business stakeholders and as Operational stakeholders.

- Vendors

The third-party vendors have many resources that you can use, and they are often ready to help if you make your needs known. They can respond quickly and are often the most cost-effective source of information and solutions.

For independent software vendor (ISV) applications that you are targeting for migration, you need to determine if the vendors provide compatible versions that support the distribution of Linux that you plan to use. Many ISV applications have other third-party dependencies. Vendors should be able to help you map out all ISV dependencies, including middleware. Most leading middleware products are available on Linux on IBM Z, and there are often open source alternatives.

- Contractors

Specialists can be called on to assist with transient needs. They may provide skills that your staff does not yet have, or skills that will not be needed after the migration project is completed. Contractors can be used to enhance the skills of your staff as they simultaneously perform tasks on the migration project. Make sure that skills transfer takes place for persistent, recurring tasks.

4.1.3 Security stakeholders

The functional area of security has become more visible and critical as company assets become more exposed to the Internet and available on mobile and wireless devices. The security stakeholders include security administrators.

The security administrators are the team responsible for data protection, including the authentication and authorization of users who access company applications. The target application must adhere to existent security policies or demonstrate heightened security methods and standards. For more details about Linux on IBM Z security, refer to 6.7, “Security analysis” on page 116.

4.2 Identify the stakeholders

The first phase of the migration involves identifying the stakeholders, as defined in section 4.1, “Stakeholder definitions” on page 50. In turn, the stakeholders identify the business and operational requirements that impact the migration process. All stakeholders within the company must be consulted to ensure that their requirements are factored into the migration planning.

As defined in section 4.1, “Stakeholder definitions” on page 50:

- Business stakeholders define the business and success criteria.
- Operational stakeholders provide information about the application requirements, database requirements, and available network bandwidth, as well as CPU load and allowable downtime.
- Security and compliance teams define compliance requirements for the entire migration effort.

4.3 Assembling the stakeholders

Holding a meeting of stakeholders (or representatives of larger groups of stakeholders) is a useful way to set expectations and to address other planning considerations. Such a meeting will help to uncover whether additional administrator, manager, or user skill enhancements are needed. The participants will also be the people to whom status and milestone results are reported. Some of these people may have never met, and a cohesive, efficient, and successful project requires personal relationships.

To make sure that all interests are taken into account, it is useful to request a meeting of the key people who requested the migration and who are affected by it. Subsets of stakeholders with related tasks and responsibilities should also meet to enhance communications and encourage teamwork.

Communicating the change

Stakeholder meetings can be an efficient way to open communication channels. Effective communications plans help to “flatten out” the negative aspects of the acceptance curve.

A communications plan, coupled with proper training on the new system, should minimize the number of users who fall into rejection or opposition mode. It encourages users to start out with acceptance instead of dissatisfaction as the initial response, and lead to a quick transition into exploration and productive use.

These issues are even more important regarding the IT support team. A strategic decision to switch an operating system or platform can inadvertently create an impression of disapproval of the work the team has done so far, and might cause staff to think their current skills are being devalued.

You should be able to articulate the objectives for your Linux migration and relate them to your key business drivers. Whether you are trying to gain efficiencies by reducing costs, increasing your flexibility, improving your ability to support and roll out new application workloads, or some other key business drivers, be sure to set up objectives that line up with these. Even the smallest of migrations should be able to do this, and it will help guide your planning.

Defining metrics (increased performance, more uptime, open standards, enterprise qualities) early in the project will help the team stay focused and reduce opposition. Be sure that you will have a means of tracking the metrics. Getting stakeholder agreement on your metrics early in the project will help ensure support ranging from executives to users.

Often, the migration to Linux will be accompanied by other objectives. For instance, some customers upgrade their database at the same time to get the latest features and performance enhancements and to obtain support that lines up well with the latest distributions of Linux. As with any project, the scope must be well defined to prevent project overrun, but it is also important that you have a means to manage additions to the plan as business needs dictate.

Because cost is often a key motivator for migrating to Linux, give careful consideration to identifying where cost reduction is targeted. Identify metrics for defining return on investment before beginning migration activities, and identify metrics for other success criteria.

4.4 Migration methodology

After the business value and need for moving to Linux on IBM Z has been accepted by the stakeholders, it is time for the actual migration planning.

In a typical migration scenario, an entire environment must be identified, rationalized, and tested for compatibility with the new host operating environment. Figure 4-1 illustrates an approach to planning.

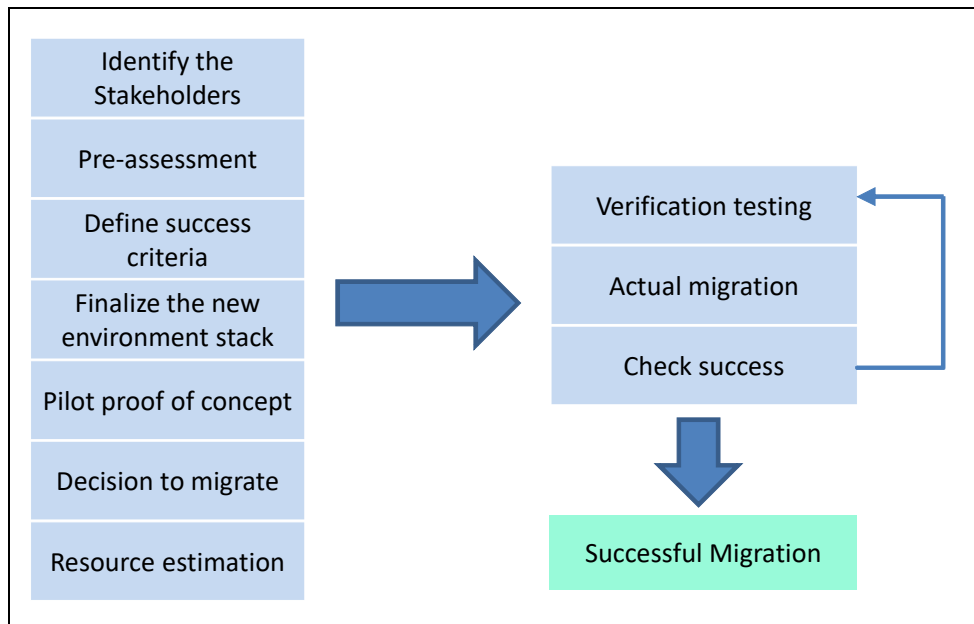


Figure 4-1 Typical migration approach

In 4.2, “Identify the stakeholders” on page 53, we discussed identifying the stakeholders and in section 4.3, “Assembling the stakeholders” on page 54. In this section, we discuss each of the remaining elements in this approach.

4.4.1 Pre-assessment

During the pre-assessment phase, a high-level analysis and initial feasibility study of the application architecture, source code dependencies, database compatibility, and build environment is performed. This task defines an overall scope for the migration to the target operating system. The applications running on current servers are assessed to determine whether they are available and certified to run on Linux on IBM Z, and an evaluation of the risks related to migration is performed. This helps to identify major risk areas at the earliest stage.

Additionally, a careful analysis of present and anticipated business needs should also be carried out and weighed against the pros and cons inherent in each option of migration. The outcome of this phase is a recommended migration approach, as well as a high-level risk assessment and analysis report identifying potential issues that can occur during the migration.

4.4.2 Define success criteria

In this phase, a consensus must be reached by all stakeholders regarding the porting project success criteria. Migration success may mean, for example, passing a percentage of system tests on the Linux on IBM Z platform or passing a level of performance criteria set out by the quality auditor in agreement with the other stakeholders.

Regardless of how the project success is defined, all stakeholders must understand and agree on the criteria before the porting effort starts. Any changes to the criteria during the course of the porting cycle must be communicated to all stakeholders and approved before replacing the existing criteria.

4.4.3 Finalizing the new environment

Usually a migration involves moving custom-built or third-party applications to another operating environment. This task involves careful analysis of different tiers of the hierarchy based on a best fit between the database, the application requirements, and other environmental attributes.

We recommend that you perform a one-to-one mapping of the various middleware, compilers, third-party tools, and their respective build parameters. If any of the one-to-one mappings for any parameters is missing, you need to list other parameters available in the tool that would provide the same functionality or feature. The 5.3, “Planning worksheets” on page 63 provides examples of forms that can be used to help document your software and hardware requirements.

During this phase, most of the technical incompatibilities and differences in the environmental options are identified and most of the times fixed.

Custom-built applications

If custom-built applications are written in one or more programming languages, several tools may need to be validated on the target environment, such as compilers, the source code management system, the build environment, and potentially third-party add-on tools.

Additionally, an in-depth analysis should be carried out on the various build options specified to ensure that the tools on the Linux on IBM Z platform provide the expected functionality after the migration (for example, static linking, library compatibilities, and other techniques). The effort involved can vary greatly depending on how portable the application code is.

ISV applications

If you are running ISV applications on x86 that you are targeting for migration, you need to determine if the vendor provides compatible versions that support the distribution and version of the target Linux on IBM Z. Many ISV applications have other third-party dependencies. Be sure to map out all ISV dependencies, including middleware. Most leading middleware products are available on Linux on IBM Z.

Note: There are many open source alternatives for many applications and services for Linux on IBM Z.

4.4.4 Pilot proof of concept

After you have a clear understanding of the target environment and the areas with possible issues and risks, you can proceed to a pilot proof of concept (PoC). This phase is a subset of the actual migration, but with a reduced scope and duration. In this phase, you implement a small module or stand-alone code snippet from the application onto the target environment.

The PoC phase should involve all of the same tasks and activities of the full migration. The main objectives of the PoC are to focus on the identified areas of risk, empirically test the recommended approaches, and prove that the full migration can be completed successfully.

In this way, the major potential migration risks identified during the pre-assessment can be addressed in a controlled environment and the optimum solution can be selected and proven. This service targets the areas of issue and risk, proves that the optimal resolution methods have been selected, and provides a minor scope of the whole migration.

Note: PoC projects may require additional funding and may lengthen the project schedule, but will likely contribute to the project's success.

4.4.5 Decision to migrate

After the pilot is complete, you should have a complete analysis of the target operating system environment as well as a roadmap detailing the resources, time, and costs required to migrate to Linux on IBM Z.

During this phase, you analyze and discuss all key requirements with the stakeholders including timing, resource needs, and business commitments such as service level agreements (SLAs). Also, discuss any related aspects of the migration, such as new workloads, infrastructure, and consolidation; the decision to implement the migration must be acceptable to all stakeholders involved in such activity, especially the business owner.

4.4.6 Resource estimation

Understanding the migration objectives and developing metrics with stakeholder involvement and agreement helps to provide a useful base from which to build a plan. Be sure to have in all key requirements (such as resource needs) and business commitments (such as service level agreements) for each stakeholder.

Migration activities rely heavily on having ready access to the personnel responsible for the development, deployment, and production support of the applications and infrastructure in question. Anticipating change and assuring the early involvement of affected teams are efficient ways to handle change issues. For example, support staff for hardware might be comfortable with UNIX-related hardware support and know where to go for help. However, practitioners who are expert in the previous environment might be less open to change if they feel threatened by new ways of doing things where they do not have expertise.

Consider the following areas in performing your resource estimation:

- Resources

Determine what hardware and software will be required. Identify the housing aspects required (for example, whether the electrical and cooling inputs are equal). Identify skill requirements. Decide what staff are needed to help with the crossover.

- ▶ Education

Identify skills-related requirements and determine whether the staff has adequate Linux and IBM Z education. Decide whether there are special skills needed for the IBM Z hardware or Linux and hardware combination.

- ▶ Service-level agreements

While installing, configuring, and testing the change is occurring, determine what the support mechanisms are for both you and any vendors. Determine what your commitments are to current stakeholders while you are performing the migration.

- ▶ Related project aspects

Be sure to check out what other projects are occurring in addition to the basic system changeover.

4.4.7 Actual migration

The scope of this phase is performing the actual migration of the applications and the infrastructure to the Linux on IBM Z environment, thus producing an environment that is ready for handover to the testing phase.

The team follows the planned approach and methodology during their migration activities. If there is a need, modifications are made to the application source code and build environment. The new application binaries are generated and checked for compliance with the target version of the operating system.

4.4.8 Verification testing

The purpose of performing a formal test is to provide objective evidence that the predefined set of test objectives is verified and the customer test requirements are validated on the target operational environment. This is an important step before verification of a successful migration. The ultimate goal is to validate the post-migration environment and confirm that all expectations have been met before committing or moving to production.

Keep the following questions in mind for validation:

- ▶ Does it interoperate correctly?
- ▶ Can it handle the expected load?
- ▶ Does it have the expected performance?

Also during this stage, if any performance issues are encountered, the target environment can be tuned for maximum performance.

4.4.9 Checking against success criteria

After you successfully migrate the environment, reassess the original acceptance criteria with all of the stakeholders. If the criteria are achieved, move the environment to production and obtain a sign-off for the migration activities. Figure 4-2 on page 59 illustrates three important criteria of success from a user perspective.

If the success criteria are not achieved, the migration implementation must be reviewed and once complete, the testing phase must be redone to ensure that the application being migrated meets the acceptance criteria and is ready to go into production.

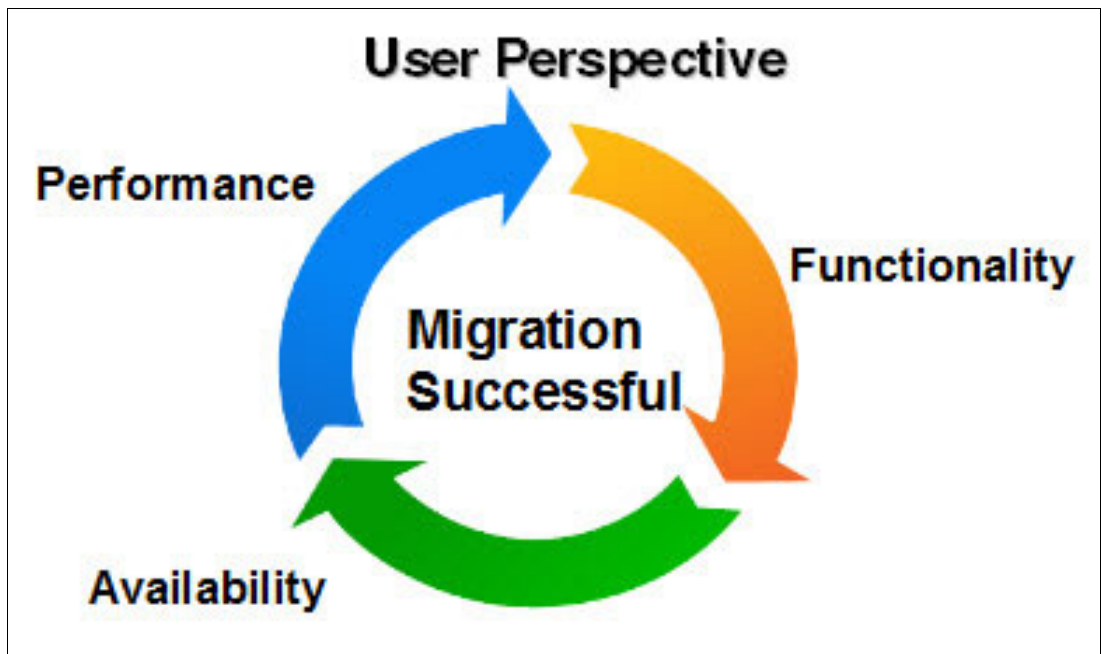


Figure 4-2 Three important criteria of success



Migration planning

This chapter provides an overview of the planning that should be done to successfully migrate workloads to Linux on IBM Z. We provide you with basic, project management information that will be useful in the planning stages of a migration project.

Additionally, we provide you with basic and generic information and templates to aid in assessing the source and target operating environments during the initial planning stages of a migration project.

The following main sections are available in this chapter:

- ▶ 5.1, “Migration project time commitments” on page 62
- ▶ 5.2, “Project definition” on page 63
- ▶ 5.3, “Planning worksheets” on page 63

5.1 Migration project time commitments

There are many phases of the migration process, and anticipating how much time might need to be scheduled for each phase may be difficult. Some practical data assembled by one of the authors of this book suggests that time spent in each of the migration activities may be distributed according to the pie chart depicted in Figure 5-1. It shows that, in general, implementation (50%) and proof of concept and testing (20%) takes most of the time. Other projects may involve more planning than implementation. Your mileage might vary, but this information may help as you prepare the migration plan, as described throughout this chapter.

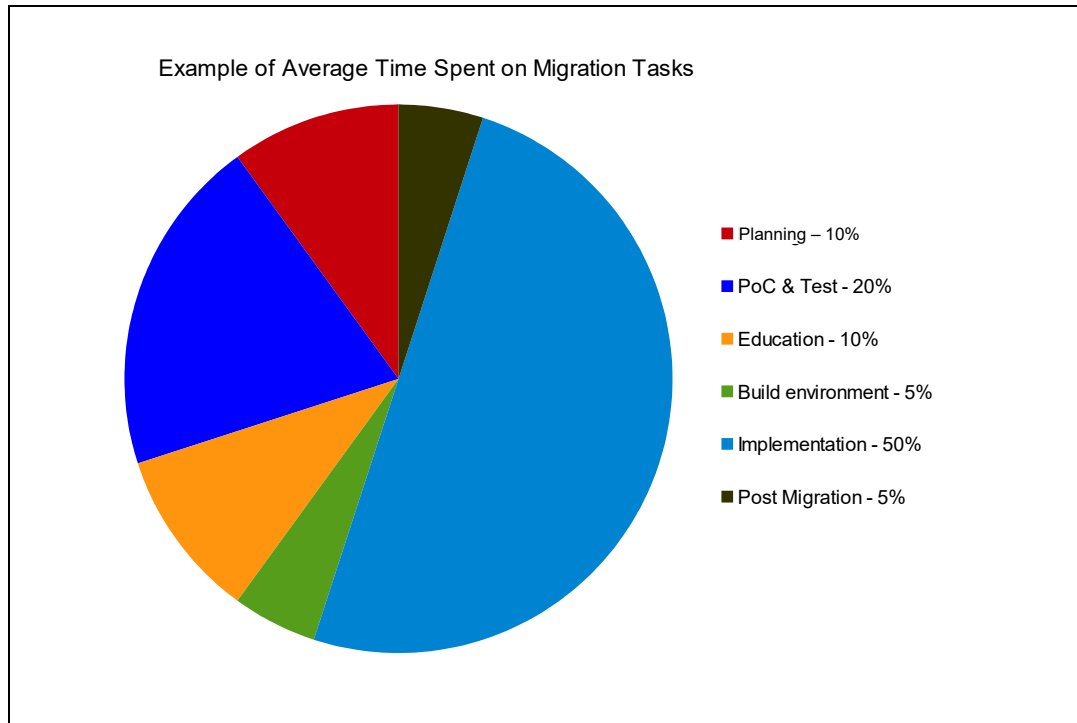


Figure 5-1 Typical time commitment averages for each aspect of migration

The phases of migration, at the highest level, can be described as:

- ▶ **Planning:** Conception of migration project plan. Once you have decided what will be migrated and how, the plan must specify the time, risks, and owner for each migration task.
- ▶ **PoC and Test:** Proof of Concept to check the compatibilities between the x86 and IBM Z environment and give special focus to performance
- ▶ **Education:** It is important that the technical staff have the right skills to work on a IBM Z migration and maintain the new environment
- ▶ **Build Environment:** In this phase, the new infrastructure will be readied for migration.
- ▶ **Implementation:** The actual migration. At this point, communication between stakeholders is important. All involved people must know and approve the migration and then have follow up reporting on the progress.
- ▶ **Postmigration:** After implementation, documentation must be created that further references the project as well as documenting all necessary maintenance and care procedures. Additionally, the project manager must have a signed acceptance agreement.

5.2 Project definition

Regardless of the size or scope of the migration, the stakeholders must start with a detailed migration plan. The success of a server consolidation project depends on several factors: Clear project definition, preparation for organizational change, sponsorship, and a good strategy. The plan gives stakeholders an understanding of the risks, savings, and deliverables, and provides an overview of the migration.

The stakeholders will discuss the project plan and produce the principal goals of the migration plan. Documents must be created that represent the strategy that will be used to accomplish the migration goals.

5.3 Planning worksheets

Planning worksheets are used to identify the hardware and software requirements, migration tasks, and project deliverables during a migration project. Although the approach and parameters for a migration planning worksheet may vary somewhat from project to project or between organizations, the foundation of an effective planning worksheet will be similar to the generic worksheets that we discuss in this chapter. The generic worksheets shown in this chapter are created for our target platform, Linux on IBM Z and can be used as a starting point for your own worksheets.

5.3.1 Software products and tooling worksheet

The software product and tools checklist template shown in Table 5-1 lists all the products and tools that are used in the source operating environment.

It provides space where you can record whether the same or similar products and tools are available on the target Linux on IBM Z operating environment.

Table 5-1 Software products and tools worksheet

Application name: Supporting business area: Technical application owner:				
Name	Version	Vendor/Source web site	License type	IBM Z

5.3.2 Application implementation worksheet

The application features worksheet dives one level deeper into the software product and tooling worksheet, where each product or tool is drilled down to their features level.

Any configurable requirements, such as specific accounts, groups, programs, or jobs that are required for an application to function should be included in this worksheet. Scenarios exist in which the same product does not offer the same features on all platforms. These details should be noted in this worksheet, as shown in Table 5-2.

Table 5-2 The application implementation worksheet

Application name: Supporting business area: Technical application owner:		
	Source (x86)	Target (Linux on IBM Z)
OS Name and Version		
Required Users		
Required Groups		
Required privileges (SUDO)		
Observations		
	Backup Solutions	
Operating System		
Database		
Hypervisor		
Container platform		
Other		
	Application-specific Dependencies	
Operating System Packages		
External Programs / Libraries		
Cron jobs		

Each product or tool listed in the product worksheet must be analyzed. All the parameters, dependencies, and optimization options must be taken into account in the source operating environment. The planning team must assess whether the same kind of features or build options are available in the target operating environment.

If the same feature is not available with the same tools or product in the target environment, the team can assess other options:

- ▶ Obtain a similar feature by linking other product or tools in the target operating environment.
- ▶ Make note of the parameters available in the same tool in the target operating environment that can be combined to give the same characteristics as in the source environment.
- ▶ If the products or product options are fully incompatible or unavailable, replacing that part of the application stack would be a useful approach to minimize the effort involved in migration. But care must be taken to ensure that all the features and parameters offered by the product in the source environment are also available in the assessing product for the target environment.

- Often, the optimization features or performance options for a product are only available for that specific platform. In such cases, the optimization features and performance options must be changed to offer the same characteristics to the target environment.

When completing the application implementation worksheet, verify whether changing parameters or options in the target operating environment has any side effects on the application or other tools used for application implementation.

If all the checklists are properly analyzed and applied, then the tools, products, and their implementation differences would be accounted for in the actual migration. This would in turn reduce the risks and the migration can be executed smoothly.

5.3.3 Application flows worksheet

The source application to be migrated can be in the center of a complex process. The application can be interconnected with many other applications, inputs, outputs, and interfaces. For this reason, you must prepare a planning document that lists the resources that the source application must provide and all the services that it is currently providing. Table 5-3 lists examples of the resources that are required of some applications.

Make the descriptions as detailed as possible by providing the physical location, server host name, IP address, network information, software product used, focal point, and any other information that you believe important to register about the services. The target environment must have the same infrastructure available to it as is available in the source environment.

Table 5-3 The application flows worksheet

Source hostname	Source IP	Target hostname	Target IP	Protocol	Port

5.3.4 Training worksheet

A critical element in achieving successful migrations is ensuring that the migration team has skills in the new technology to be migrated. Ensure that a training checklist is put into place during the planning process. Identify the people to be trained, the skills that need to be imparted and a timetable of when the training needs to be done to ensure that staff are trained at the right time.

5.3.5 Hardware planning worksheet

The hardware planning worksheet lists the hardware resources that you must consider during a migration project. The source system resources are examined and mapped to a similar or more advanced technology that is on IBM Z. An example of how to complete this process is shown in Table 5-4 on page 66.

Table 5-4 Example of a completed hardware planning worksheet

HARDWARE PLANING WORKSHEET			
SERVER NAME:			
RESOURCE	SOURCE	DESTINATION	OBSERVATION
Number of CPU	4	2	Real to Virtual
System memory (in GB)	8	8	
OS SWAP Memory (in GB)	4	4x512M	Disk to VDISK
Network connection^a			
Connection Description	Gigabit Ethernet	Gigabit Ethernet	
Connection Type	Gigabit Ethernet	vSwitch/GbE	
IP Address/Netmask	129.40.19.88/24	129.40.23.153/24	
VLAN number: vSwitch	2	2 : vSwitch1	
Disk Resource^b			
OS File system	/ : 30 : Ext3	/ : 2 :Ext4	Root
Mount Point: Size (in GB): Type		/home : 3 :Ext4 VG OS	Logical Volume
Mount Point: Size (in GB): Type		/opt : 5 :Ext4 VG OS	Logical Volume
Mount Point: Size (in GB): Type		/tmp : 5 :Ext4 VG OS	Logical Volume
Mount Point: Size (in GB): Type		/var : 1 :Ext4 VG OS	Logical Volume
DATA File system			
Mount Point: Size (in GB): Type	/DB : 100 : Ext3	/DB:100:XFS VG DB	Logical Volume
Mount Point: Size (in GB): Type	/WAS : 50 : Ext3	/WAS:50:XFS VG DATA	Logical Volume
CUSTOM File system			
Mount Point : Size (in GB) : Type		/MGM:10:BTRFS VG DATA	Logical Volume
Logical Volumes: Volume Group OS : 20GB Volume Group DB : 150GB Volume Group WAS: 80GB Volume Group MGM: 20GB			

a. The following network connections are available for IBM Z:

- Ethernet/QETH
- Open vSwitch
- IBM HiperSockets
- Direct OSA-Express connection

b. Logical Volume Manager (LVM) provides storage management flexibility and reduced downtime with online resizing of logical volumes

5.3.6 Firewall planning checklist

Firewalls are a critical component of a security architecture. Make sure you document the network flows in order to avoid problems and reduce errors during the migration phase.

The firewall planning checklist lists the firewall rules that you need to consider during a migration project. In the checklist used in this project, the Table 5-5 can be used as template to help you identify which ports will be used by each application, what type of protocol will be used and the source and destination for the network flow.

Table 5-5 Firewall checklist table

Source IP/name	Destination IP/Name	Traffic type (TCP, UDP)	Destination Port	Description

Take the opportunity to review all network flows and ensure it is in compliance with your organization security policies.

5.3.7 Security and privacy worksheet

Whether you're managing personal data or protecting critical business assets, implementing robust security and privacy practices is essential. This comprehensive worksheet serves as a tool to assess your current approach and identify areas for improvement:

- ▶ Assess the effectiveness of your current security measures and identify any potential weaknesses.
- ▶ Pinpoint specific areas where your security and privacy practices could be strengthened.
- ▶ Formulate a road map for addressing identified shortcomings and bolstering your overall data protection strategy.

These details should be noted in this worksheet, as shown in Table 5-6.

Table 5-6 Security and privacy worksheet

Security worksheet			
Control	Description	Implemented (Yes/No)	Notes
Physical Security			
Secure physical access to servers and network equipment	(for example, locked doors, security cameras)		
Regular security checks of physical infrastructure			
Inventory of equipment and software			
Disposal of outdated or unused equipment in a secure manner			
Data Security			
Strong password policies for user accounts	(for example, minimum length, complexity requirements)		

Security worksheet			
User access controls	(for example, least privilege principle, role-based access)		
Encryption of sensitive data at rest and in transit	(for example, procedures for identifying, containing, and recovering from security incidents)		
Vulnerability management program	(for example, regular patching of software, vulnerability scanning)		
Incident response plan			
Regular backups of data with off-site storage			
Data loss prevention (DLP) controls	(for example, to prevent unauthorized data transfer)		
Network Security			
Firewall to control network traffic			
Intrusion detection/prevention system (IDS/IPS)			
Secure configuration of network devices			
Monitoring of network activity for suspicious behavior			
Data collection and use			
Do you have a privacy policy that clearly explains what data you collect, how it is used, and with whom it is shared?			
Do you obtain user consent before collecting personal data?			
Do you provide users with the ability to access and update their personal data?			
Do you have procedures in place to securely dispose of personal data when it is no longer needed?			
Third-Party data sharing			
Do you share user data with any third parties? If so, who are they and what data do you share?			
Do you have contracts in place with third parties that ensure they protect user data in accordance with applicable privacy laws?			

Security worksheet			
Do you allow users to opt-out of having their data shared with third parties?			

By using this worksheet, you can leverage security and privacy features provided by IBM Z such as pervasive encryption, technology-enforced (rather than administrator-enforced) isolation of workloads at massive scale, the use of Trusted Platform Module, compliance with privacy regulations and protecting sensitive data with quantum-safe cryptography.

5.3.8 Sustainability worksheet

Use the worksheet shown in Table 5-7 to assess your current server environment and plan for sustainable workload consolidation on Linux on IBM Z.

Table 5-7 Sustainability worksheet

Task	Completed (Yes/No)	Notes
Planning and Assessment		
Analyzed current server resource utilization		
Assessed total server environment energy consumption		
Estimated future workload growth		
Implementation and Optimization		
Selected energy-efficient LinuxONE models		
Implemented virtualization technologies		
Established plan for monitoring and optimizing resource usage		
Identified and partnered with e-waste disposal company		
Reporting and Review		
Established method for tracking and reporting energy savings		
Scheduled regular review and update of sustainability plan		

The sustainability worksheet can be used to document any additional considerations or challenges related to sustainable workload consolidation.

Note: Consider seeking expert advice from IBM representative or sustainability specialists for guidance throughout the planning and implementation process.

Use the following tools to track energy consumption for your existing x86 servers:

- Operating system-level tools:

- Microsoft Windows:
 - Performance Monitor is a built-in tool providing detailed power consumption data for various components.
 - Power Efficiency Diagnostics Report is a tool that analyzes system settings and provides recommendations for energy optimization.
- Linux:
 - Use a tool like [PowerTop](#) to monitor and analyze real-time power consumption, and get insights into resource usage and optimization opportunities.
- ▶ Hardware-based tools
 - Management Information Base (MIB): Embedded in some servers, allowing access to detailed power consumption data through SNMP (Simple Network Management Protocol) tools.
 - Intelligent Platform Management Interface (IPMI): Hardware interface providing detailed power consumption data and remote management capabilities for some servers.
- ▶ Third-party Software:
 - Several vendor-specific or general-purpose IT monitoring solutions offer advanced power consumption tracking capabilities, often integrated with other system health and performance metrics.

You can use the following tools to track energy consumption for your IBM Z servers:

- ▶ Hardware Management Console: With IBM z16, a new [Environmental Dashboard](#) experience has replaced the Energy-Efficiency Statistics task on the HMC. This new experience allows for:
 - Monitoring the system and partition power consumption.
 - Selecting time ranges and metrics to view historical data.
 - The ability to chart and analyze the data.
 - Ability to export the data for use elsewhere.
- ▶ HMC Web Services API: Energy management data, aligned with the ASHRAE Tier 1 classification, can be accessed by external DCIM systems through a secure, REST-based Web Services API
- ▶ Optimizing for sustainability with IBM Z
 - Being able to prove the energy efficiency of consolidation on IBM Z is important. This is core to IBM's sustainability software strategy - to advise, inform, and connect the board-room to every layer of operations. Through this integration with IBM Instana®, Grafana and IBM Envizi™, you can show the benefits of optimizing the power envelope of your workloads by migrating them from x86 to IBM Z.
 - For more information about IBM Envizi ESG Suite, see [IBM Envizi ESG Suite](#).
- ▶ [Instana zHMC Sensor](#): track your energy and resource utilization, alongside your application performance, using Instana Observability.
- ▶ [TCO and CO2e Calculator](#): Simple but powerful online (public) calculator that estimates the TCO of a specific system including its environmental (CO2e) aspects.
- ▶ [Power Estimation Tool](#): Tool that estimates the power consumption, line cord phase currents, system weight, airflow and exhaust air temperature for the specified configuration. New for the rack mount offering is the power supply current and power consumption of the components.



Migration analysis

This chapter helps you to understand new features that you will find on Linux on IBM Z and provide a technical direction for your migration. Each section addresses a different part of your infrastructure using scenarios to exemplify how the migration will affect the environment.

The following main sections are available in this chapter:

- ▶ 6.1, “Network analysis” on page 72
- ▶ 6.2, “Storage analysis” on page 84
- ▶ 6.3, “VMware to KVM migration options” on page 94
- ▶ 6.5, “Database analysis” on page 105
- ▶ 6.6, “Backup analysis” on page 112
- ▶ 6.7, “Security analysis” on page 116
- ▶ 6.8, “Operational analysis” on page 127
- ▶ 6.9, “Disaster recovery and availability analysis” on page 130

6.1 Network analysis

This section provides information about network migration configuration issues, explains how the virtual network can be configured, and the facilities available on IBM Z and IBM z/VM.

6.1.1 Network facilities available on IBM Z and z/VM

On the mainframe, different network devices are available for use. Many of these come from a historical background, and should not be used for new implementations. They commonly stay, however, to continue the support of previous installations on newer hardware. Linux on IBM Z can operate using all common network interfaces but for new installations, there are recommended methods for operation depending on the use case.

The following are some technologies that you will find in the IBM Z world that are not used or even seen on x86 systems. This section clarifies some new facilities that you are going to find when you are migrating from x86 to IBM Z. We provide some brief information that you can use to start your network planning. In each subsection, you can find a reference for more detailed information.

Open Systems Adapter

The Open Systems Adapter Express (OSA-Express) is a hardware network controller. It is installed in a mainframe I/O drawer and provides connectivity to clients on local area networks (LANs) or wide area networks (WANs). It can be directly attached on Linux, but in z/VM they will typically be attached to virtual switches (see “Virtual switch” on page 74). You can find more technical information about OSA cards on [IBM z16 \(3931\) Technical Guide, SG24-8951](#).

OSA with Link Aggregation

You can aggregate multiple physical OSA cards into a single logical link, which is called a link aggregation group (LAG). This configuration increases the bandwidth and provides nondisruptive failover.

Note: Link Aggregation is supported by OSA Express adapters, but the function is actually performed by the operating system. Importantly, the z/OS operating system does not have link aggregation capability.

In z/VM, Link Aggregation is performed by the z/VM Virtual Switch (including the Global vSwitch). In Linux, the bonding kernel module provides a number of link aggregation modes, such as Link Aggregation Control Protocol (LACP) and round-robin.

For more information, see [Advanced Networking Concepts Applied Using Linux on IBM System z, SG24-7995](#).

HiperSockets

HiperSockets is a microcode implementation of an Ethernet-like network. LPARs on an IBM Z server can access a HiperSockets network through an emulated interface similar to an OSA-Express adapter. HiperSockets provides communications with near zero latency at memory speed between servers running in different LPARs. HiperSockets must be configured in the I/O configuration of the mainframe.

HiperSockets provide a very fast connection between LPARs. They provide an easy way to connect many Linux servers to a z/OS system in the same mainframe.

This direct connection without involving real hardware is an important factor to simplify setups with many Linux systems that must be connected to z/OS. Some benefits are explained in [Set up Linux on IBM System z for Production, SG24-8137](#).

z/VM HiperSockets-vSwitch Bridge

HiperSockets networks are isolated from any external interfaces and do not provide external connections. Normally, if an external connection is required, either the Linux guest must have an additional interface to attach to a vSwitch or other external connection, or another Linux guest must be set up as a router.

The z/VM HiperSockets vSwitch Bridge creates a connection between HiperSockets and a z/VM vSwitch to allow systems attached to a HiperSockets network to access external network resources. For a Linux guest attached to a HiperSockets network, it now does not need another external interface or a router guest to access external resources. The z/VM HiperSockets-vSwitch Bridge will take any network packets from the HiperSockets-attached system that are not intended for another system on the HiperSockets network and send them out through the vSwitch uplink.

Note: The z/VM HiperSockets-vSwitch Bridge does **not** bridge traffic between guests attached to the vSwitch and systems on the HiperSockets network. For guests attached to the vSwitch, their traffic path is **always** through the vSwitch uplink port, even if the destination is on the HiperSockets network.

The z/VM HiperSockets-vSwitch Bridge is the z/VM portion of the overall approach to simplification of high-speed low-latency communication between Linux under z/VM and z/OS. The z/OS side of the solution is discussed in the following paragraphs.

HiperSockets Converged Interface (HSCI)

When connecting a Linux guest to an IBM z/OS system on the mainframe, the HiperSockets network in Layer 3 mode was the method to use. From the perspective of the TCP/IP configuration, the HiperSockets network is a separate TCP/IP network that introduces potential issues around routing and multi-homing.

To alleviate these issues and to increase HiperSockets adoption, the z/OS HiperSockets Converged Interface (HSCI) was introduced in z/OS V2R3. Instead of being a separate IP network with separate addressing, the HiperSockets network operates as an extension of the external network that z/OS is attached to.

When HSCI is configured, TCP/IP creates a logical association between a HiperSockets network and one or more OSA-attached networks. This association is defined in the IBM Z hardware configuration, using an attribute called the physical network ID (PNETID). When z/OS Communications Server has to send a packet, it checks to see whether the destination is on the HiperSockets network, and if it is, it sends the packet on the HiperSockets network instead of the OSA-Express.

To use HSCI, a HiperSockets network with a particular attribute is needed. In HCD/HCM the “External Bridge” attribute provides the required support¹. If you are already using HiperSockets and want to start using HSCI, we recommend to define a new HiperSockets network for HSCI use.

HSCI can greatly simplify the configuration of environments with extensive connectivity between Linux (running under z/VM or in LPAR) and z/OS, when combined with the z/VM HiperSockets-vSwitch Bridge (discussed in the previous section).

¹ In DPM, HiperSockets networks have their required attributes set automatically. At the time of writing, there is no support for running z/OS within a DPM-mode system. This is merely an observation, given the limitation.

However, z/OS installations may need to do a review of their existing connectivity to ensure that HSCl can be accommodated in their HA configuration.

RDMA over Converged Ethernet (RoCE)

RDMA (Remote Direct Memory Access) over Converged Ethernet (RoCE) is a feature that allows for high-speed, low-latency networking on IBM Z. It utilizes the RoCE Express hardware feature to make connections to other Linux on Z and z/OS instances on other physical machines. This technology helps to reduce the CPU consumption of TCP/IP when data is transmitted over its links. More information is provided in [Networking with RoCE Express, SC34-7745-00](#).

Virtual switch

A virtual switch (vSwitch) is a software program that enables one virtual host to communicate with another virtual host within a computer system. Virtual switches typically emulate functions of a physical Ethernet switch. In Linux on IBM Z, a vSwitch provides direct attachment of z/VM guests to the local physical network segment. The vSwitch allows IP network architects and network administrators to treat z/VM guests as a server in the network.

The switched network inside a z/VM Operating System commonly is implemented with a vSwitch. When running the vSwitch as Layer 2, it behaves similar to a real switch just between virtual machines.

The actual speed of a connection with a vSwitch depends on a number of different variables. The type of traffic is as important as the real underlying hardware and the maximum transmission unit (MTU), which is the maximum size (in bytes) of one packet of data that can be transferred in a network. Common to all of those solutions is that the vSwitch is faster than a real switch connected to the mainframe would be.

vSwitches do not need a connection to an OSA card to operate. They can also provide purely virtual networks. This also simplifies the setup of private interconnects between guest systems. When creating private interconnects in an SSI with LGR enabled, the use of dedicated VLANs with external interface is recommended. This is necessary to accomplish the private connection between guests that run on different nodes in the SSI.

Implementing VLANs also helps if different guests run in different security zones of a network. It is easy to configure network interfaces to Linux guests that provide only selected VLANs to the guest. These can be configured either as tagged VLANs or as a single untagged VLAN on an interface.

The vSwitch infrastructure provides two basic configuration options. One configures user-based access, the other configures port-based access. From the possibilities, both are equivalent, just the configurations differs.

You can read more about vSwitch benefits on [Set up Linux on IBM System z for Production, SG24-8137](#), and technical information about [Advanced Networking Concepts Applied Using Linux on IBM System z, SG24-7995](#).

6.1.2 Network migration overview

There are several different levels of network migration that should be considered because z/VM provides a complete virtual network system, which includes the ability to create multiple virtual switches in the same LPAR. vSwitches allow, among other features, the use of VLANs.

The vSwitch operates at either Layer 2 or Layer 3 of the OSI Reference Model, and is virtually attached to the same network segment where the OSA card is physically connected.

In this section, we show some common scenarios and how they look on IBM Z.

Single network scenario

One of the most common scenarios is the migration of several distributed machines from the same physical subnet to a single IBM Z LPAR attached to the same network segment.

Figure 6-1 shows an example depicting a single distributed network.

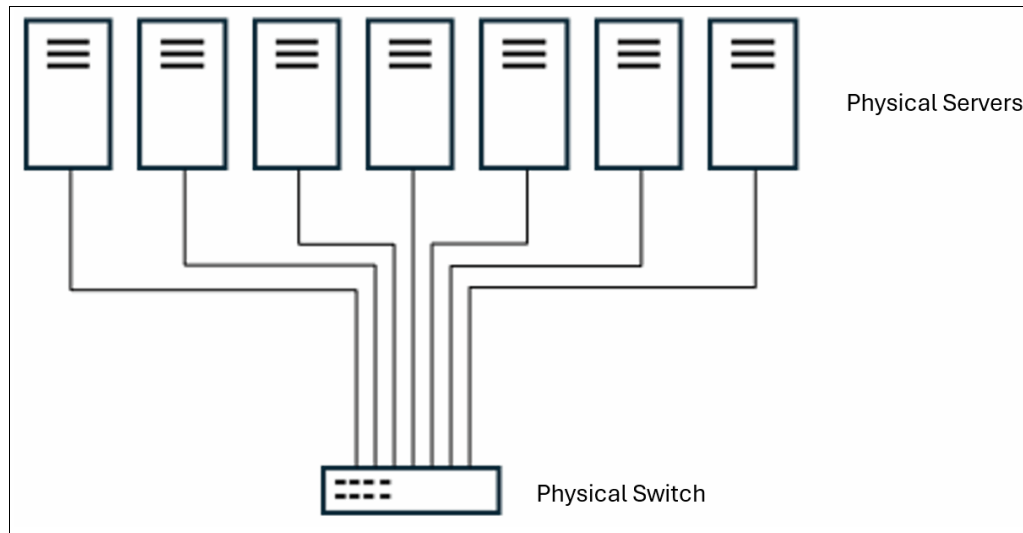


Figure 6-1 Single distributed network

Within this scenario, all physical machines can be migrated to a single IBM Z machine running Linux and sharing the same vSwitch, which is attached to an OSA card. The OSA card is then connected to the physical network. Figure 6-2 on page 76 illustrates this type of configuration.

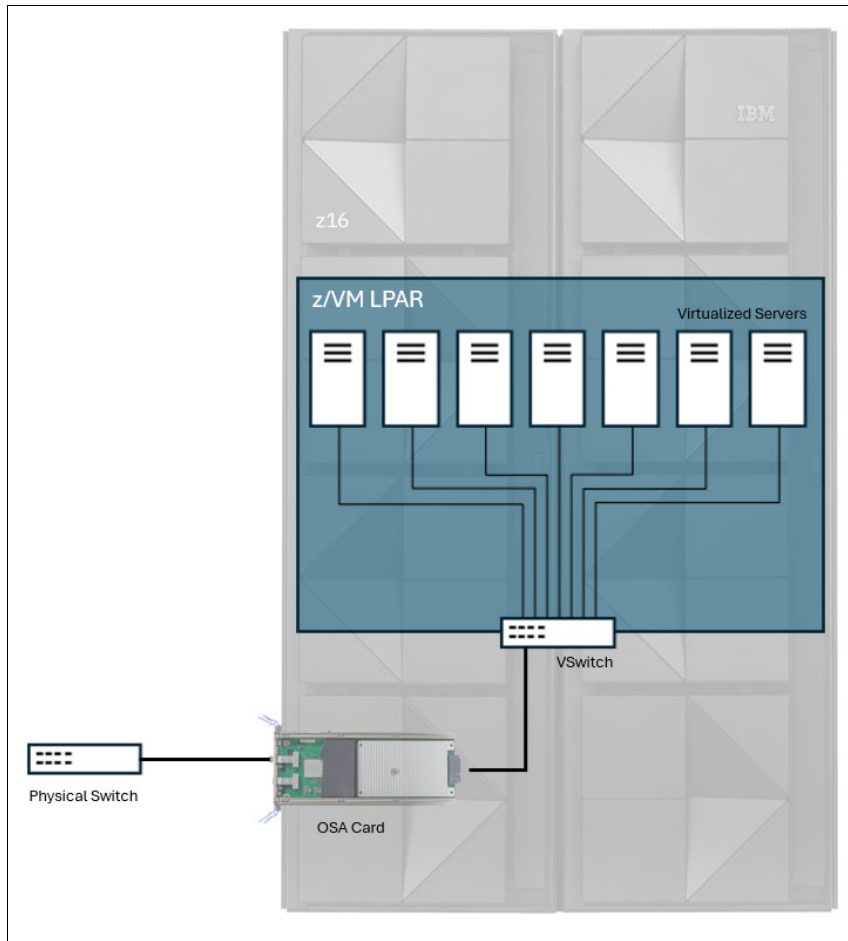


Figure 6-2 Single virtualized network

To increase the availability of each Linux guest, the recommended solution is to configure two or three OSA cards attached to different physical switches in the network. This provides a network failover capability, as illustrated in Figure 6-3 on page 77.

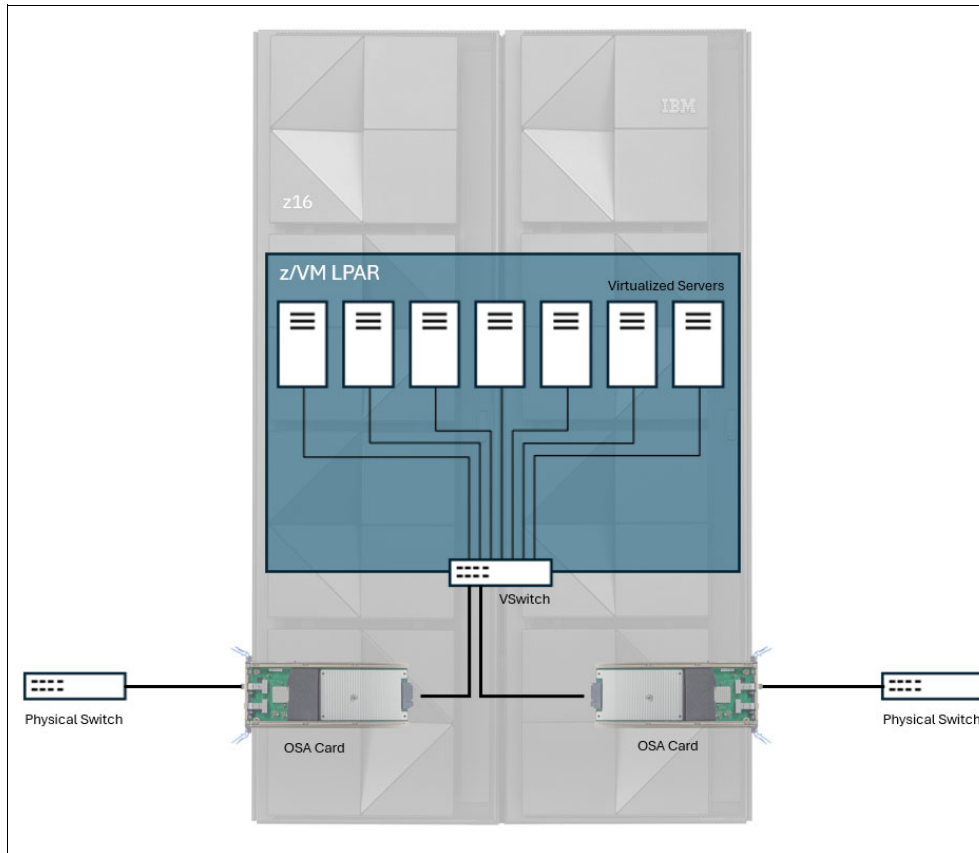


Figure 6-3 Single virtualized network with failover solution

In a Layer 2 vSwitch configuration, all Linux guests have their own media access control (MAC) address. In a Layer 3 vSwitch configuration, the Linux guests respond with the OSA card's MAC address to requests from outside the IBM Z LAN segment.

In a multiple LPAR scenario where a single network segment is used, the recommended solution is to share the OSA card between LPARs. Each LPAR's vSwitch is connected to the OSA card and the OSA card is directly connected to the physical network segment. This is a common scenario where the development and production server are in separate LPARs. This configuration is illustrated in Figure 6-4 on page 78.

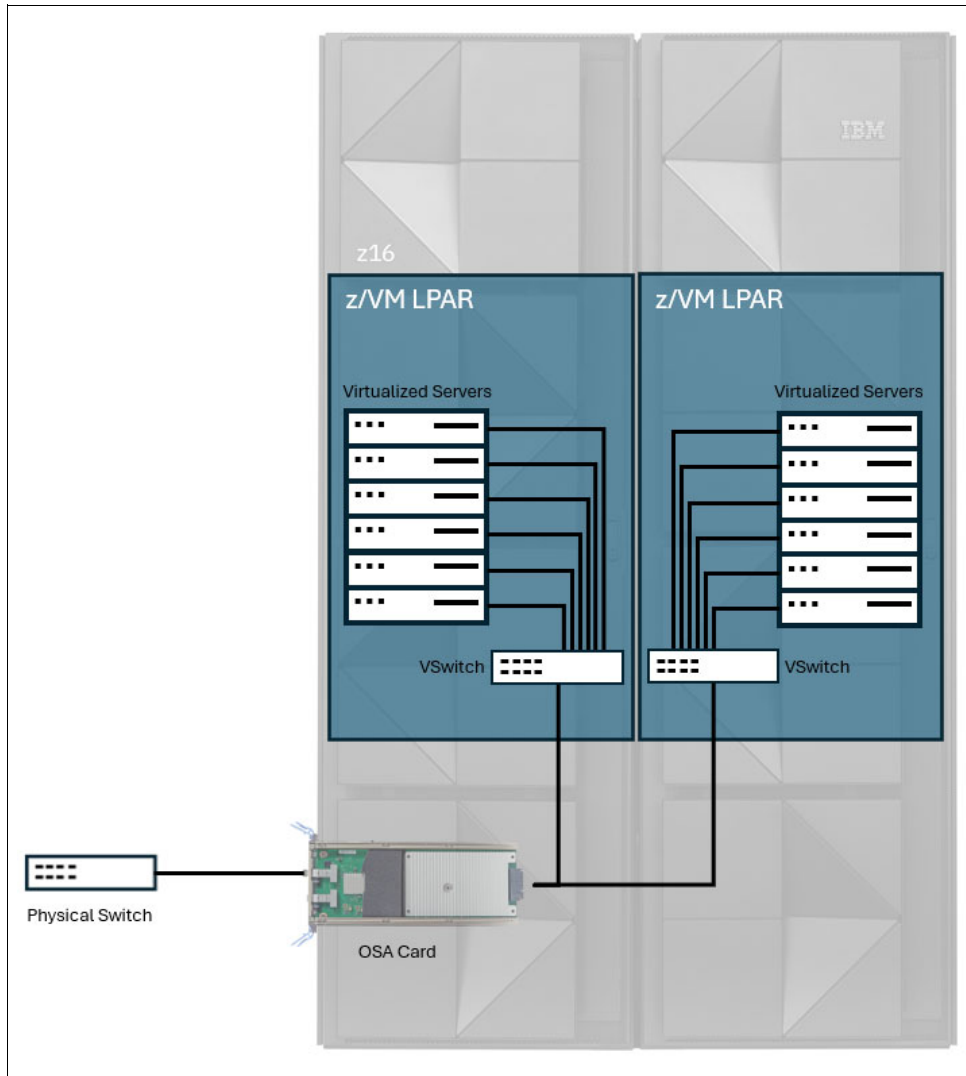


Figure 6-4 Single virtualized network with multiple LPARs

Similarly, the failover solution described previously can also be applied in this case. Sharing the two OSA cards between LPARs is shown in Figure 6-5 on page 79.

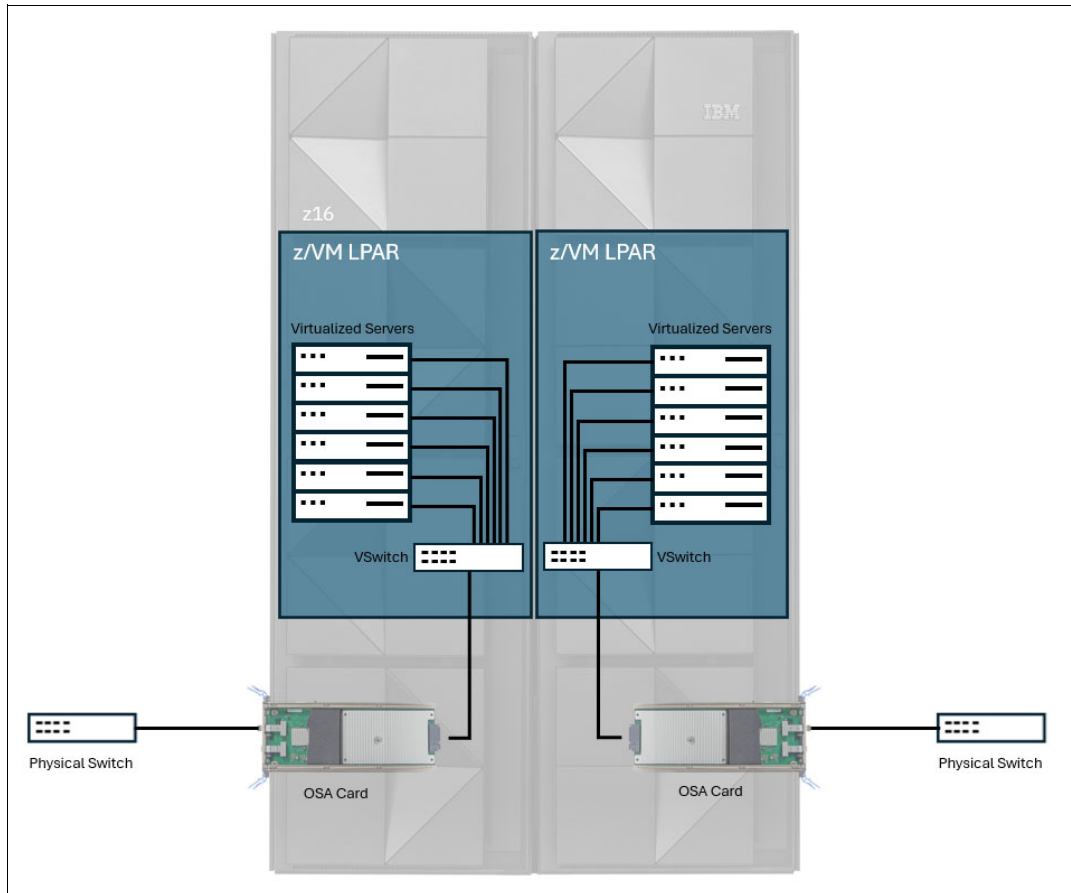


Figure 6-5 Single virtualized network with multiple LPARs and failover

Multiple network scenario

There are several types of network solutions that require multiple network segments. Some of these demand package routing or the use of multiple LPARs. This section provides suggestions for each type of network design.

DMZ and secure network

In some scenarios, different network segments are migrated to Linux on System z and share the same IBM Z. We are analyzing the demilitarized zone, or DMZ and a secure network scenario. Figure 6-6 on page 80 shows a DMZ network where the Web Application Server is placed, and a secure network where the database server is located.

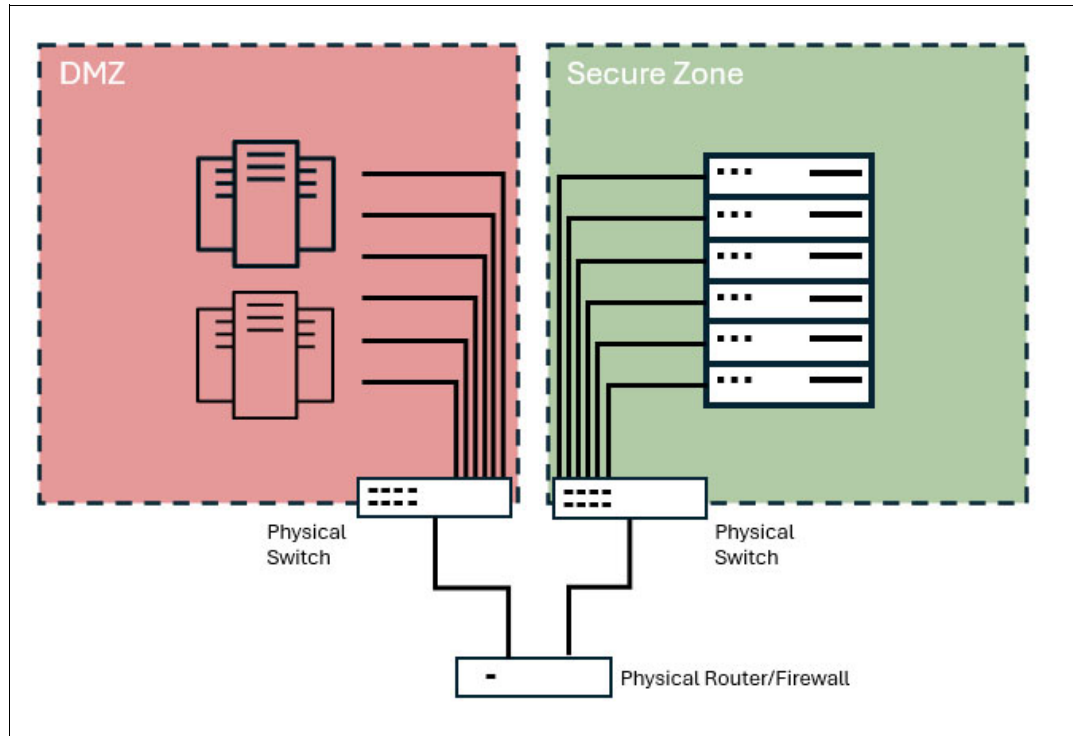


Figure 6-6 Multiple distributed network scenario: DMZ segmented network

You can set up the same scenario on IBM Z. If you have in place a physical switch, a third-party firewall solution, and a router in your environment, you can reuse them as part of your network planning on IBM Z. Otherwise, you can use some network facilities available on z/VM and IBM Z.

The OSA card is connected to one physical switch (or two OSA cards, when the failover solution is configured). The physical firewall can be replaced by a Linux guest that can act as a router and firewall (if you do not have an appliance firewall solution). All virtual Linux guests will be connected to two vSwitches setting two different network segments. Figure 6-7 on page 81 shows a network using a Linux guest as a firewall.

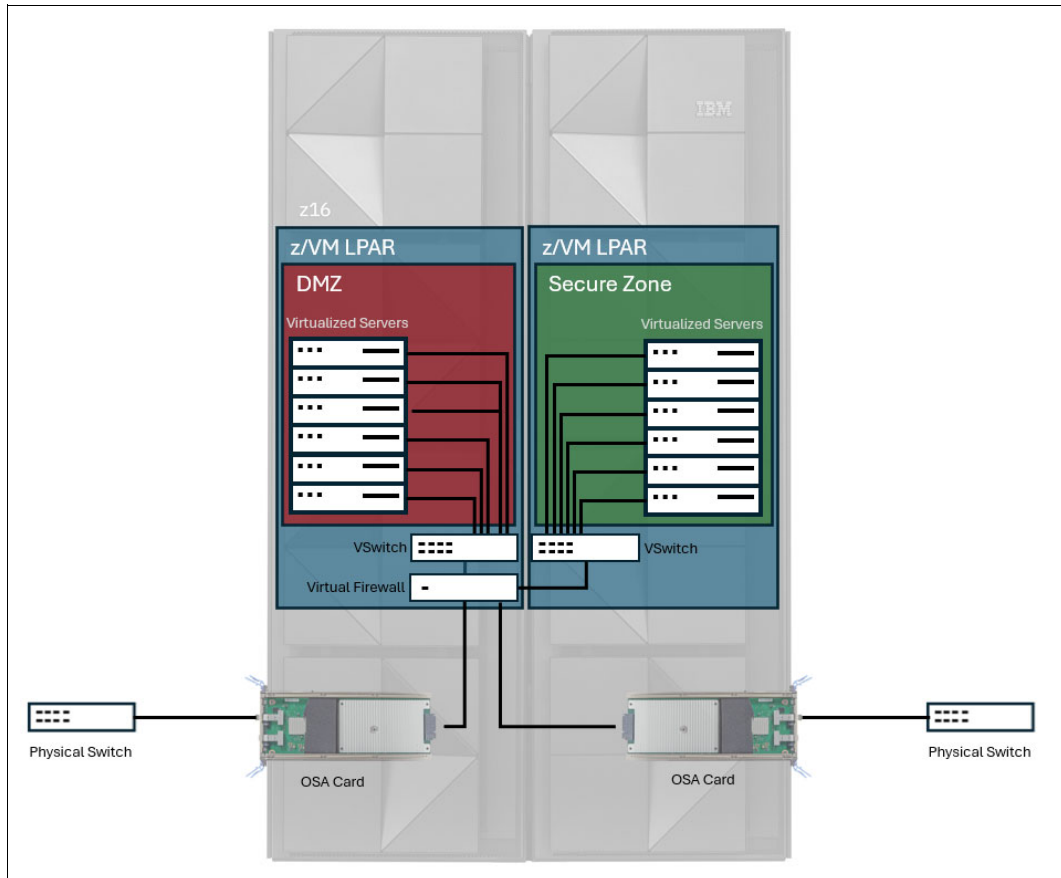


Figure 6-7 Multiple virtualized network scenario: DMZ and secure network

You might have noticed in Figure 6-7 that we are not sharing the OSA cards. It is possible to have the OSA card shared between multiple LPARs on the same IBM Z hardware. To create this solution, it is recommended that you have an external firewall to manage the network filters. Figure 6-8 on page 82 illustrates the solution that is described as a network segmented LPAR.

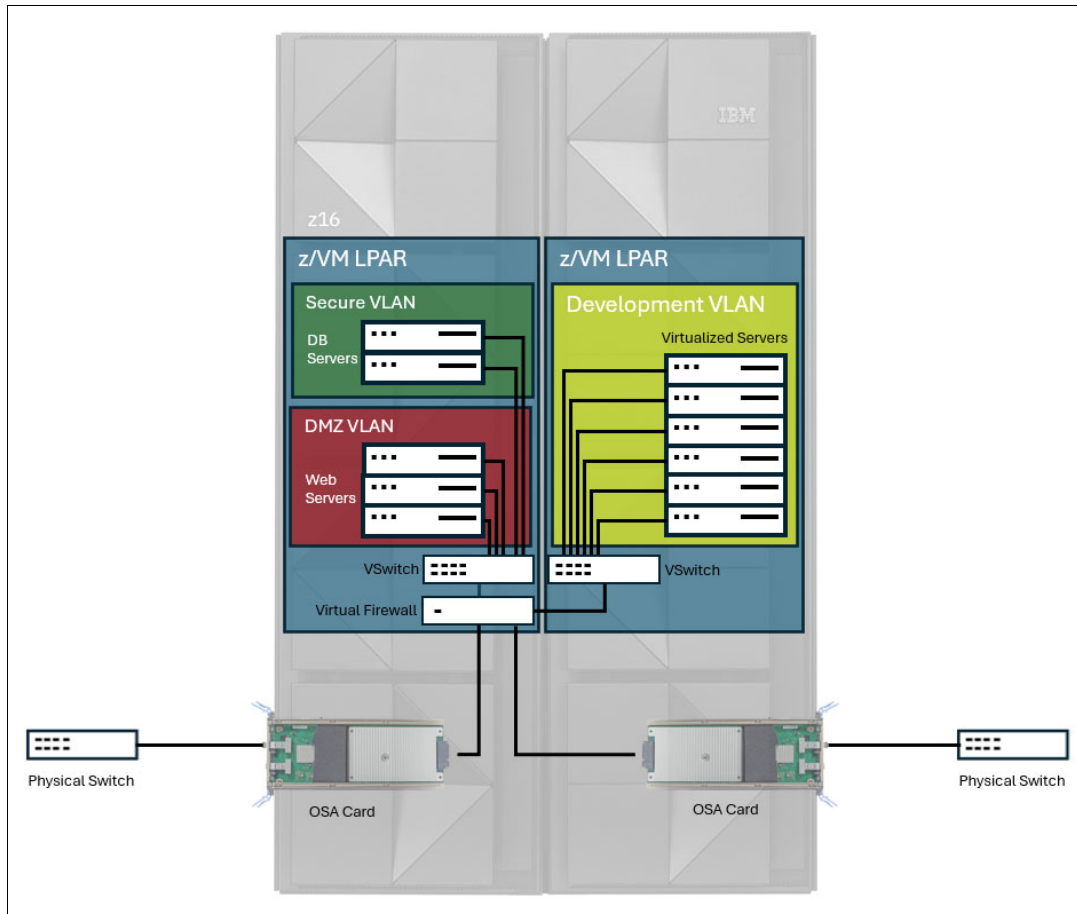


Figure 6-8 Multiple virtualized network scenario with failover: DMZ and secure network

You can isolate the entire secure network from the physical network segment using multiple LPARs. The communication between the LPARs is managed by HiperSockets devices.

Figure 6-9 on page 83 illustrates an example.

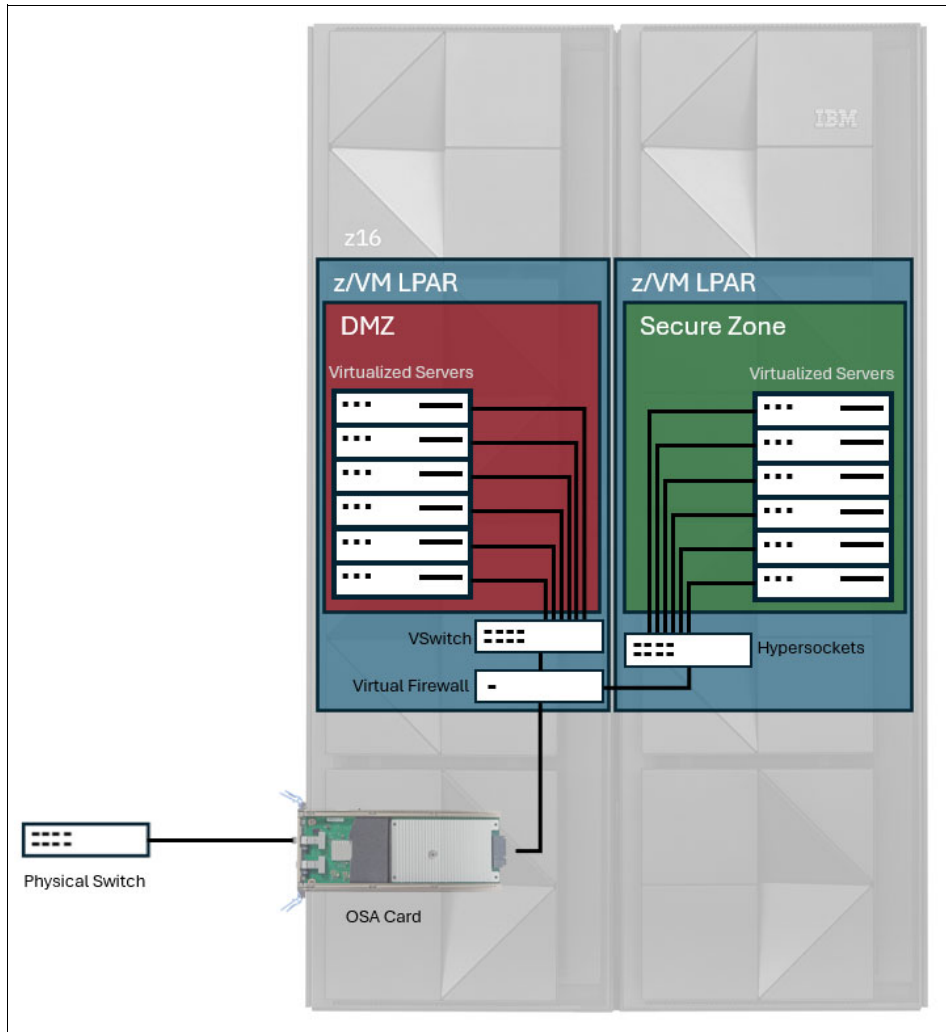


Figure 6-9 Multiple virtualized network scenario with multiples LPARs

Note: Although the use of HyperSockets for this scenario is possible, it might not be the recommended solution. If one of the LPARs is CPU-constrained, that could cause a delay of network traffic. You can read more about HyperSockets in *Set up Linux on IBM System z for Production*, SG24-8137.

VLAN segmented network

The use of the VLAN tag on z/VM VSWITCHes is fully supported. The VLAN configuration will help in the segmentation of network packages, bringing security and organization to the environment. It will facilitate the administration of network grouping the guests with common requirements regardless of their physical location. The vSwitch, like a physical switch, will provide full authorization on a per port basis for membership in a VLAN segment.

For a high security network scenario, use the LPAR environment mixed with the multiple network segmented solution. As illustrated in Figure 6-8 on page 82, the entire IBM Z environment is virtualized and all configurations are made per virtual machine, which increases the security, reduces the complexity, and simplifies the environment.

6.1.3 Helpful steps for a network migration

The Linux on IBM Z administrators and network administrators should work together to engineer the best solution for your environment. Here are the basic steps:

1. Determine the new IP address for the new servers. The IP address should be on the same IP network to minimize the number of variables of the entire migration.
2. Determine the VLAN IDs of the Linux on IBM Z servers.
3. Configure the vSwitch with the listed VLAN IDs.
4. Configure the Linux servers using the designated IP addresses.

At this point, the target server (Linux on IBM Z server) must be assigned a host name that is different from the source server name:

1. Migrate the applications (for more information, see section 6.3, “VMware to KVM migration options” on page 94) and files from the source server to the target server.
2. Shut down the source server.
3. Change the Linux on IBM Z server’s host name.
4. Change the DNS registered name to the new Linux on IBM Z IP address.

If the application running is an IP-based application, it is possible to change the IP address of the target Linux on IBM Z server to the source IP address.

6.2 Storage analysis

This section explains concepts and designs, such as online migration and offline migration, regarding the storage configuration possibilities for Linux on IBM Z. Other storage migration issues are also covered.

6.2.1 Data migration

Two models of data migration are discussed in this section: online migration and offline migration:

- ▶ Online migration refers to the case where the source server, target servers, and all services are up and running and a system outage is not required.
- ▶ Offline migration requires a service outage to switch over from the source server to the target servers.

We examine both migration models in more detail in the following subsections.

In both types of data migration, some unexpected issues must carefully be considered. The result of not doing so could lead to an extended outage or unexpected downtime, data corruption, missing data, or data loss.

Online data migration

Some applications are eligible for online migration. To be eligible, basically, an application must provide multi-operating system clustering support and be available on Linux on IBM Z.

To perform an online migration, follow these steps:

1. Install and configure the target Linux on IBM Z server. For more details, see 6.2.2, “Linux on IBM Z: pre-installation considerations” on page 88.
2. Install the middleware application on the Linux on IBM Z server.
3. Copy the application data to the target Linux on IBM Z server.

The software application selection depends on the type of data that needs to be copied. Solutions like the Linux **scp** program can be used in online data migrations where the application does not change or the changes are totally controlled.

Otherwise, the Rsync software application can be used to synchronize the application data between the server in a small period of time during the migration process.

4. Include the Linux on IBM Z server in a cluster as a cluster node.
5. Monitor the Linux on IBM Z server to verify that the application is responding to requests correctly.

This step is not a test of the application on Linux on IBM Z. The application must be tested on a development machine to guarantee that the application is a Linux on IBM Z compatible application (refer to 6.3, “VMware to KVM migration options” on page 94 for more details).

6. Shut down the source servers.

Always consider the content of the data that is migrated before choosing online migrations as a solution.

To avoid such issues, online data migration must always be executed during off-hours, and you should always take a data backup just before the actual data migration activity begins.

Offline data migration

Offline data migration can apply to all system migrations. This kind of data migration can be accomplished by using several different approaches and functionality including:

- ▶ Using the network mount points NFS or Samba connections and either the **DD** or **CP** Linux command.
- ▶ Using an FTP server on the source or target server.
- ▶ Using an SCP/SSH server between the server and the target server.
- ▶ Using the Rsync synchronization application between the source or target server.
- ▶ Attaching the storage volume to a Fibre Channel device (Linux-to-Linux migration).

Using the Rsync application

For a better result using the Rsync application, schedule service synchronization for an entire week before the outage by following these steps:

1. On the first migration day, execute the first synchronization.

Execute the first synchronization during a time when the use of the server is low. (Rsync only copies files that are not locked, thereby avoiding any issues with files in use.) During this time, however, server response might be slower than normal because of the extra read I/O activity.

2. During the migration week, you can execute a daily synchronization at the server during off-peak hours.

Only modified files will be copied from the source to the target server.

3. The last synchronization day is the server outage day, when access to the source server is denied to users.

Because there are no open files, the Rsync application will be able to copy all files to the target servers.

4. Shut down the source servers and start all services on the target Linux on IBM Z servers.

Transferring files over the network

Database migrations are the most common example of the requirement for files to be transferred over the network. That is because most database software needs an offline backup that includes a data export or data dump to a new file.

That exported/dumped file needs to be transferred across the network, and the database import procedure must be executed at the target server. Refer to 6.5, “Database analysis” on page 105 for more details.

Migrating storage volumes

When the source servers are Linux x86 connected to an external storage device using Fibre Channel, and if there is a zFCP device that is part of the same storage area network, it is possible to connect the source Linux volume to the target Linux server on IBM Z. However, both servers cannot share the same volume.

Storage SAN Volume Controller

One option available to simplify the storage and data migration for Fibre Channel disks involved in a migration to Linux on IBM Z is to install the IBM System Storage SAN Volume Controller.

The SAN Volume Controller sits in the channel path and allows you to virtualize all FCP storage from multiple vendors that sit behind it. Figure 6-10 on page 87 shows where the SAN Volume Controller sits in the storage area network (SAN). The SAN Volume Controller has visibility to all supported storage on the SAN.

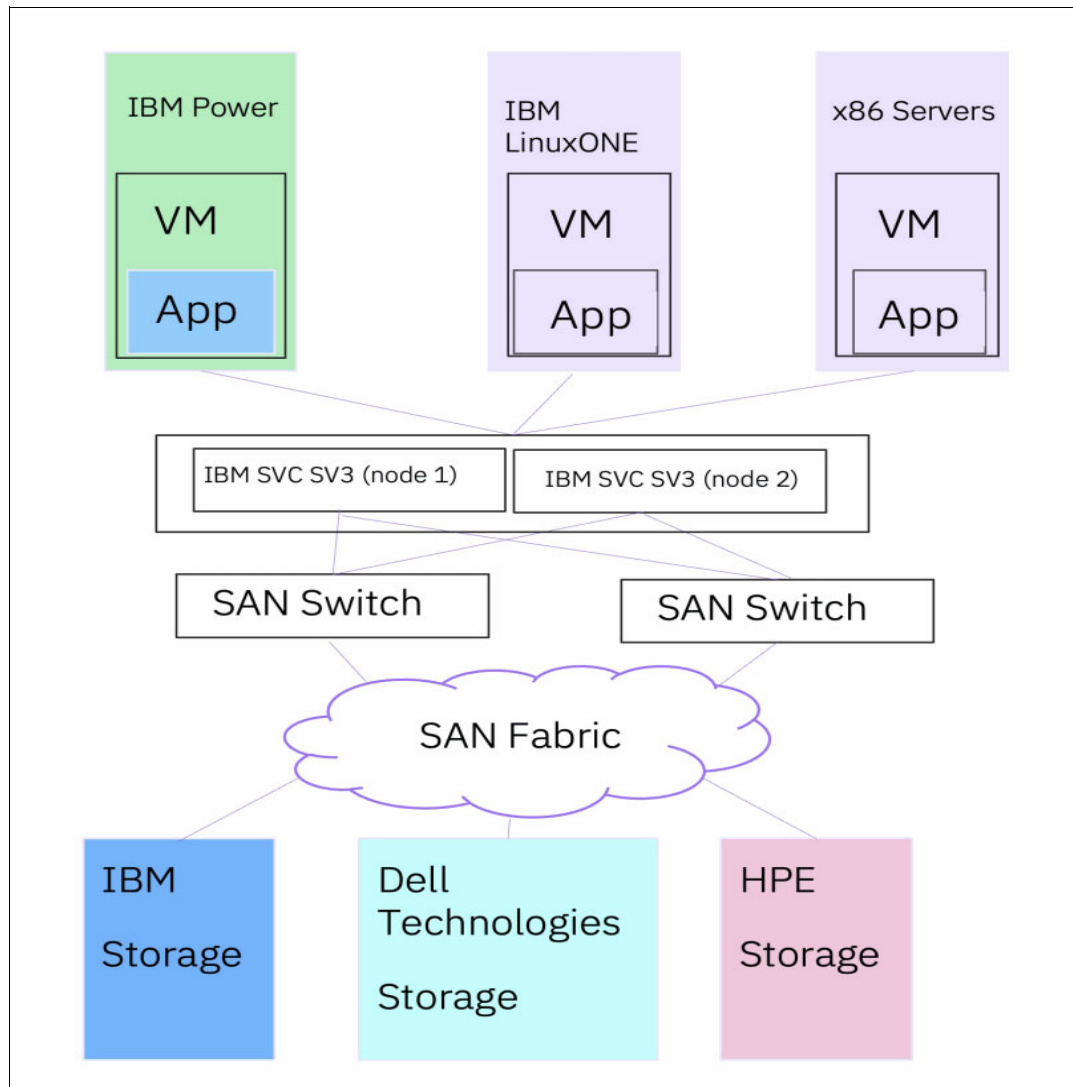


Figure 6-10 SAN Volume Controller

The following benefits are provided by the SVC:

- ▶ Single point of control for heterogeneous storage resources
- ▶ Dynamic data migration between heterogeneous storage devices on a SAN
- ▶ Ability to pool the storage capacity of multiple storage systems on a SAN
- ▶ Scalability to support up to 1024 host servers
- ▶ Instant copies of data across multiple storage systems with IBM FlashCopy®. For more information, see [IBM Storage FlashSystem](#).
- ▶ Copy data across metropolitan and global distances as needed to create high-availability storage solutions. For more details on this topic, see [Implementing the IBM System Storage SAN Volume Controller with IBM Spectrum Virtualize V8.2.1](#), SG24-7933.

When migrating Linux systems from x86 to Linux on IBM Z, the SAN Volume Controller will allow you to non-disruptively migrate data to Linux on IBM Z. For more information about the IBM System Storage SAN Volume Controller, see [IBM SAN Volume Controller](#).

- ▶ Additional information can be found in the following publications:

- [Introduction to Storage Area Networks, SG24-5470](#)
- [Implementation Guide for IBM Storage FlashSystem and IBM SAN Volume Controller \(for IBM Storage Virtualize 8.6\)](#)
- [Performance and Best Practices Guide for IBM Spectrum Virtualize 8.5, SG24-8521](#)
- [IBM SAN Volume Controller Model SV3 Product Guide \(for IBM Storage Virtualize V8.6\), REDP-5670](#)
- [IBM SAN Volume Controller Best Practices and Performance Guidelines, SG24-8502](#)

Note: If you are using [IBM Storage Virtualize with FlashWatch](#) program you can benefit from free data migration for 90 days to seamlessly transfer your data from over 500 different storage controllers, regardless of brand (IBM or non-IBM).

Helpful steps for an auxiliary storage migration

The multiple possibilities provided by Linux on IBM Z to store and access files lead to many types of solutions. The solution you architect for the target system will dramatically affect the flexibility, efficiency, and performance of the migrated application.

For source applications that reside on servers where storage is local or the external storage is not compatible with Fibre Channel data storage, all data must be copied using the network file system from the source server to the target server (Linux on IBM Z):

1. Create the new server file system with mount points for all data files.
2. Create a temporary file system to be used in the file transfer process on the target server.
3. Configure the target server as an NFS file server, a Samba file server, or an FTP File Server to upload the files from the source server.

Note the following points:

- If there is enough space at the source server to compact all of the data, consider using data compression features such as **zip**, or **tar** with **gzip** and **bzip** formats. Both of these formats are compatible with Linux on IBM Z. The data can be transferred using an FTP server configured on the target server.
 - If there is not enough space at the source server to compact the data, mount the NFS file system or map the Samba file system at the source machine, and copy the files across the network.
4. Verify the correct files permissions at the target directory. Adjust file permissions after the transfers for production work.

For file storage in an external storage system compatible with Fibre Channel, we can migrate to a Linux on IBM Z server configured with zFCP adapters to connect directly to the volumes that should be migrated to Linux on IBM Z servers.

6.2.2 Linux on IBM Z: pre-installation considerations

The storage and file system design has a direct influence on system performance, system availability, and the capabilities for system expansion.

A best practice for Linux on IBM Z is that only one version of a Linux OS distribution should be installed from scratch. Therefore, the basic Linux file system should be designed to allow the highest possible model of servers and then all other Linux guests in the environment should be cloned from this source (known as the golden image).

The file system that stores the application data is created after the cloning process depending on the needs of the application that will reside on the server. If you want to know how to create a SLES 11 or RHEL 6.4 golden image, see [The Virtualization Cookbook for IBM Z Volume 1: IBM z/VM 7.2, SG24-8147](#).

Logical Volume Manager (LVM)

All file systems, except the root (/) file system, should be created as LVM devices. File systems created with an LVM will make it possible to expand or reduce the file without a system outage (using SLES 10 SP2 or higher, or RHEL 5.0 or higher and LVM2).

The Logical Volume Manager (LVM) is very useful for Linux file systems because it allows you to dynamically manage file system size and has tools to help back up and restore failing partitions.

Basically, LVM volumes are composed of the following components:

- Physical volume

A physical volume (PV) is a storage device, and it can be a DASD device or a SCSI device controlled by a zFCP channel. For Linux on IBM Z, each DASD device is a physical volume.

- Logical volume

A logical volume (LV) is the disk partition of the LVM system. This is the area that is formatted and is accessed by users and applications. The LV is exposed through a mount point.

- Volume group

A volume group (VG) is the highest level of the LVM unit. A volume group is created by one or more physical volumes and gathers together the logical volumes.

Figure 6-11 on page 90 shows five minidisk (MDisk) devices that are used by a Linux guest to create a unique VG. It is then further organized or allocated into two LVs.

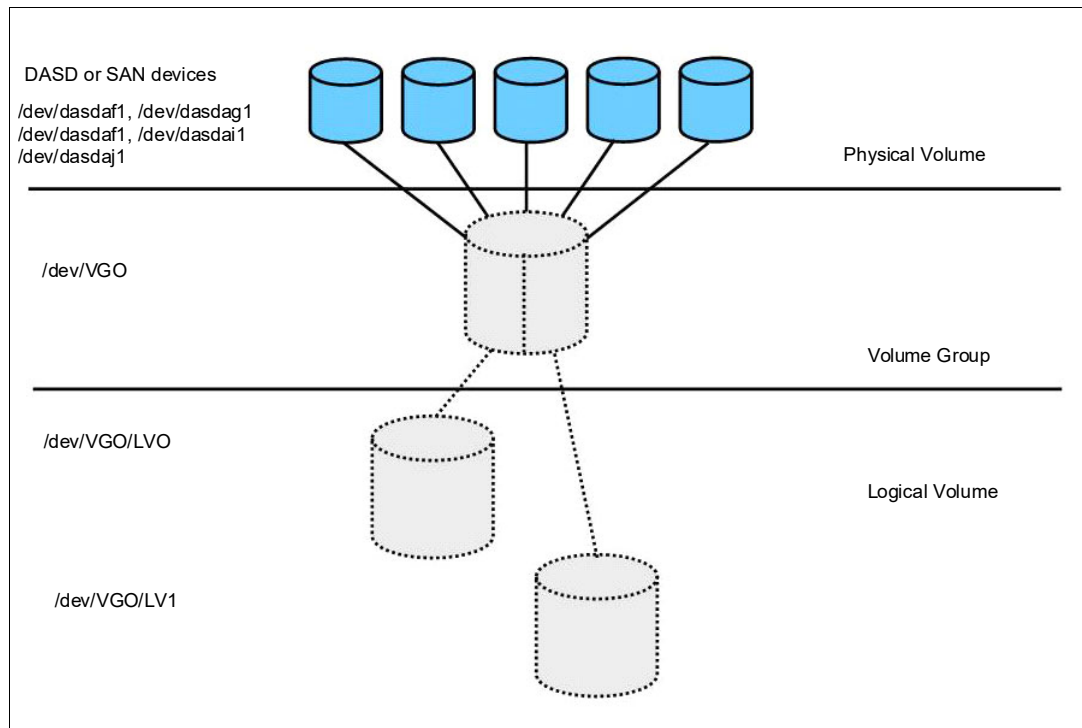


Figure 6-11 LVM example

There is, however, a small performance price that must be paid when using LVM. The flexibility of LVM often outweighs the cost of the performance hit.

For more information about the LVM setup during the installation, see [The Virtualization Cookbook for IBM Z Volume 1: IBM z/VM 7.2, SG24-8147](#).

Linux file system

As mentioned previously, the basic Linux OS file system should be designed so that one single image (the golden image or prototype) can be cloned to be used on as many Linux servers as possible.

The golden image should include the following file systems:

- ▶ root (/) file system
- ▶ /boot file system
- ▶ /usr file system
- ▶ /var file system
- ▶ /tmp file system
- ▶ /opt file system
- ▶ /home file system

In the following sections, we discuss these file systems in more detail.

The root (/) file system

The root file system is the first file system to be created, and it is the base for all other file systems in the hierarchical structures of the Linux operating system. A size of 10 GB is the minimum requirement for RHEL.

Important: The root (/) file system should not be placed on an LVM device because in case of an LVM failure, you can recover the system using the single user mode.

The /boot file system

The /boot file system is often left simply as a subdirectory under root (/), but maintaining this directory structure as its own partition can be particularly useful. /boot contains the boot files, such as the kernel, the parm file, the initial ramdisk, and the system map. In SUSE Enterprise Linux Server and RHEL, the /boot partition also contains the boot loader configurations, such as zipl or GRUB. As it holds the kernel files, it may be considered to be the most important partition of all. Keeping it as its own partition helps preserve its important status and maintain its integrity.

Important: Like root (/), the /boot file system should not be placed on an LVM device. The recommended file system type for /boot is EXT3.

The /usr file system

The /usr file system is where all Linux standard base applications are installed. The binaries, libraries, and shared files are copied to this directory during the installation process. The file system size depends on the type of server you are running and on the distribution-based packages that need to be installed for the functions that the server provides.

The golden image /usr file system size should be the minimum to support the basic Linux distribution files. The ability to increase this file system is necessary because after cloning the server, the system administrator might need to increase the file system to install new packages or additional package dependencies.

This file system should be created on LVM devices that allow you to dynamically extend or reduce the file system size.

In a shared Linux on IBM Z environment, this file system could be set as read-only because the system simply needs to read the application file into memory. This also offers an added security benefit because no one can delete or change any file in a directory mounted as read-only.

The /var file system

The /var file system is where all the variables files (such as spool files, cache files, and log files) are written. The /var file system has files that are constantly changing such as /var/log/messages and /var/log/secure.

The size of this file system depends on the number and type of applications that are running and how long the log files will be kept on the server. Also, consider whether the application is designed to write files here, as well as their sizes and frequencies.

The services control files are also placed on the /var file system so it could never be scaled to be a shared file system and it must be always read/write.

Because it is a dynamic file system, it should be placed on an LVM device to allow it to be extended or reduced as needed.

The /tmp file system

The /tmp file system was originally designed to store operating system and temporary application files that would be deleted every time that system is rebooted or deleted by the application right after the file is no longer in use. Some homemade applications use the /tmp

file system as a dump area or an exchange file resource. In rare cases, the size of the /tmp will need to be increased.

Because it is a dynamic file system, it should be placed on an LVM device to allow the capability to be extended or reduced as needed.

The /opt file system

The /opt file system is where all third-party applications should be deployed. As a best practice, the /opt directory should be further organized by the company or organization that developed the application or software. The next directory level would be to specify the software package that is installed. For example, a Db2 for Linux server should be installed at /opt/ibm/db2. A WebSphere Application Server should be placed in the /opt/ibm/WebSphere directory.

The file system size will depend upon the size of the software packages that will be installed in it. It is easy to estimate the requirements for a single software package. But upgrades, maintenance, and additional software packages are not so easy to plan for. The /opt file system can also be a dynamic file system and should be configured on an LVM device.

The /home file system

The /home file system is designed to allocate user files. The size of the file system will depend upon the server function and the number of users defined on the server. For example, application production servers do not need a large /home file system because it is not expected that development staff will store files on a production server. However, it is expected that applications will be developed on a development application server, so developers will need sufficient file system space to create and transfer their files.

Depending upon the situation, the /home file system could be a dynamic file system. If it is dynamic, it should be configured on an LVM device.

Other file systems

An example of additional file systems that could be created on a specific server during the migration process is the database server file system. Basically, you need to have at least one file system for data files and one for log files. Therefore, at a minimum two file systems should be created in addition to the file system where the application binary files would be installed. For an IBM Db2 database server, the default location for the binary files is /opt/ibm/DB2.

Other database management systems put their data files in other directories. For example, the MySQL database server default location for data files is the /var/lib/mysql directory. If the server is a MySQL database server and you are using the Linux distribution from Red Hat Linux or SUSE Linux, consider including a new file system at the /var/lib/mysql mount point.

For each target database management server, make sure that you know where the binary files and the data files will be located, because only then can you plan to create the devices and file systems for the target system.

It is possible that there are file location differences depending upon the distribution of Linux that you install at your site. Make sure that you know these differences, if any, and plan for them.

Additional resource

You can see additional recommendations, like volume group and disk naming convention in [Set up Linux on IBM System z for Production, SG24-8137](#).

Shared file system

The data storage in a Linux on IBM Z environment can be shared physically by one or more Linux guests. However, because of limitations of the file system, it is not possible for two Linux guests to have read/write control to a device at the same time, although z/VM allows it at the hardware level.

In a shared DASD environment, keep in mind that the file system changes performed by the guest machine that has the read/write control will only be available to all other guests that share the same file system after unmount and mount of the file system. As an example, think of the environment of a web cluster service where the application servers only need read access to the web pages and do not need to write to the same file system where the files are allocated.

In the example shown in Figure 6-12, only the special file system and mount points relevant to the solution are represented. The data file location is at mount point `/srv/www/app`. This is the file system that is shared between the Linux guests. There is also the shared file system `/opt/ibm/IBMHTTPD`, where the web server binaries are installed. For the IBM HTTP service, the log files are redirected to the local `/var/log/httpd` file system. All shared devices are the same DASD device type and managed by the z/VM operating system.

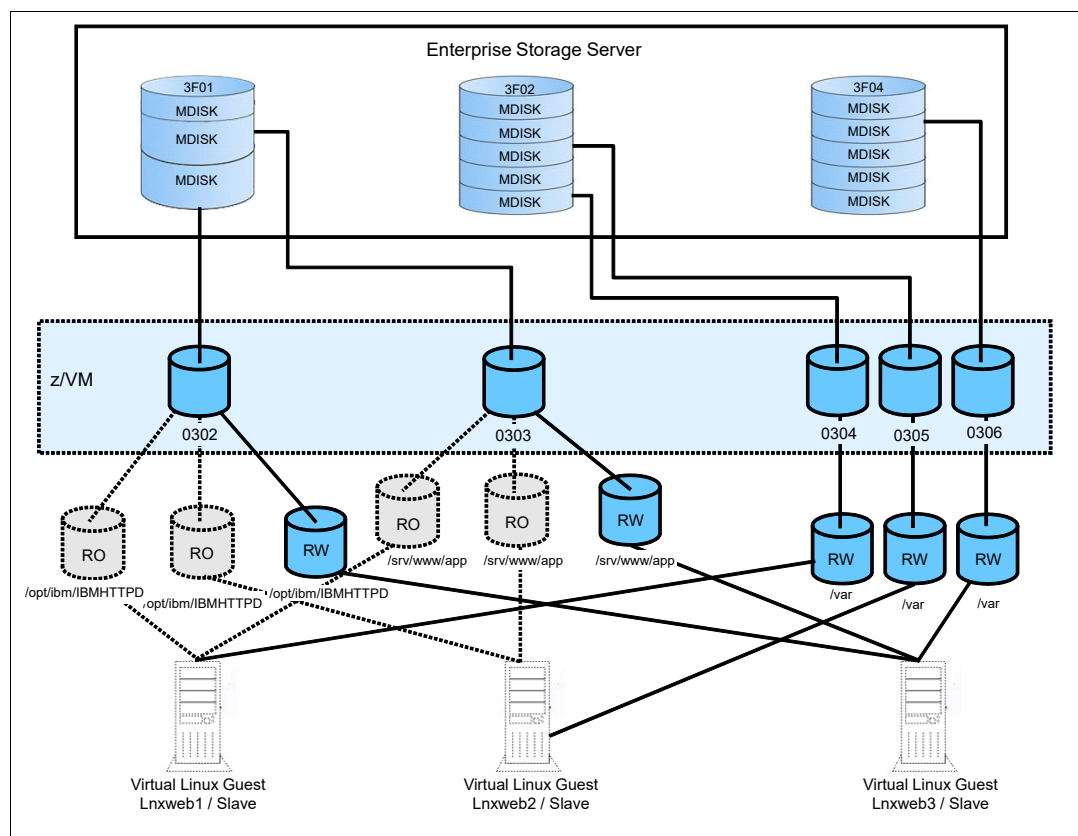


Figure 6-12 Shared devices example

The benefits of using a shared file system are based on economy of resource. You can reduce application binary space allocation and code updating efforts because you only have to update one master server and just remount it on the subordinate servers.

Note: System administrators must pay special attention to managing this kind of environment because if the same file system is mounted as read/write in two different servers, all data can be lost.

ECKD and zFCP devices

ECKD and zFCP devices can be shared by the same Linux guest. This is a common and helpful approach when using large file systems, as in the case of database servers.

The zFCP device, when configured with multiple access channels, provides a better I/O response than a single ECKD channel device. After it is configured on Linux on IBM Z, it is possible to split it into partitions like a simple SCSI device using the **FDISK** tool. Even though the sizes of the ECKD volume devices are determined at the storage hardware level, it is still possible to configure smaller volume sizes for the Linux guest when the z/VM system administrator formats the ECKD devices as MDisk devices.

A combination of both solutions can help you improve system performance and use storage resources efficiently. For more information, see [Linux on IBM System z: Performance Measurement and Tuning, SG24-6926](#).

6.3 VMware to KVM migration options

There are several tools available to convert from VMware to KVM such as **virt-v2v** and OpenStack (after exporting the OVF/OVA file using VMware tools or commands).

Note: virt-v2v is compatible only with the AMD64 and Intel 64 architectures (x86_64). Other architectures, such as IBM Z, IBM POWER®, and 64-bit ARM, are not supported for v2v conversion.

Evaluating the migration of the mission-critical applications from VMware to the KVM must include careful balancing of pros and cons. Often times it is not a good idea to directly going through the conversion process as the best solution:

- ▶ Mission-critical applications often have dependencies and configurations within the VMware environment. Attempting to convert them directly to KVM can introduce complexity and potential risks to the stability and performance of these applications.
- ▶ VMware and KVM have different internal architecture and sets of features which can make the conversion process incompatible. Specific VMware-provided features may not have “one-to-one” equivalents in KVM, and in such cases, it becomes necessary to deal with extra configuration or customization to make certain that the migrated applications can function as expected.
- ▶ Data integrity and security assurance is therefore one of the issues that should be kept in mind, more so in the case of mission critical applications. Actually, the whole migration procedure opens a possible place for data corruption or exposure unless it is managed properly.

Instead of VMware to KVM conversion, it is advisable to opt for a fresh deployment and installation of the applications on the KVM hypervisor. This approach offers several advantages:

- ▶ Starting with a clean slate allows for a fresh setup tailored to the KVM hypervisor, optimizing performance and resource utilization. It eliminates any existing dependencies or configurations that may no longer be relevant or efficient.

- ▶ Each new deployment is an opportunity to tune the system and artifacts of these applications so that they work optimally with KVM. This can be realized by utilizing the hypervisor-specific features and optimizations in order to boost up performance, scalability and reliability.
- ▶ Different middleware, such as Oracle, WebSphere, IBM Db2® and the others, provide features that overcome platform specific applications by making migration possible. These tools can simplify the process of migration to achieve a smooth transition and along the line minimize errors. Frequently, they involve having built-in testing automation, verification, and rollback functionalities which will reduce the risks related to migration.
- ▶ When utilizing containers, one of the recommended best practices is to adopt a multi-architecture image build approach. This strategy ensures that your containerized applications can be deployed across various architectures (x86, S390, ppc64le...) enhancing compatibility across different infrastructure environments.

6.4 Application analysis

This section describes the analysis you need to perform to identify applications that would be good candidates for migrating to Linux on IBM Z.

We discuss the following topics:

- ▶ How to identify the best candidates for a migration to Linux on IBM Z.
- ▶ How to select the appropriate application for a Linux on IBM Z proof of concept.
- ▶ What you can do if your ISV does not support Linux on IBM Z.
- ▶ How you can accommodate application interdependencies in a Linux on IBM Z environment.
- ▶ How you can redesign your application to take advantage of the strengths of the IBM Z platform.

6.4.1 Application architecture overview

A common way to ease the selection of migration candidates to IBM Z typically involves a comprehensive architecture analysis of each application in your portfolio. It often consists of an architecture diagram that provides an abstraction of many aspects of an application.

An application's infrastructure diagram helps you to understand the relationship among its interconnected components and assess the overall migration complexity. With a diagram available, it is possible to fully establish expectations, required efforts, and goals along with all involved stakeholders during the migration process, which typically speeds up the migration process.

Figure 6-13 on page 96 shows an application's infrastructure architecture. Consider the following points:

- ▶ Five Linux servers are necessary.
- ▶ The WebSphere Application Servers and Db2 servers must be in the same network zone.
- ▶ The IBM MQ and Node.js systems must be in a restricted network zone.
- ▶ Only traffic from the WebSphere servers to the MQ server must be permitted across the two network zones.

After the initial application's architecture assessment is complete and it is considered a candidate for migration, all of the stakeholders should have a much deeper understanding about what the tasks that are necessary to achieve a successful migration.

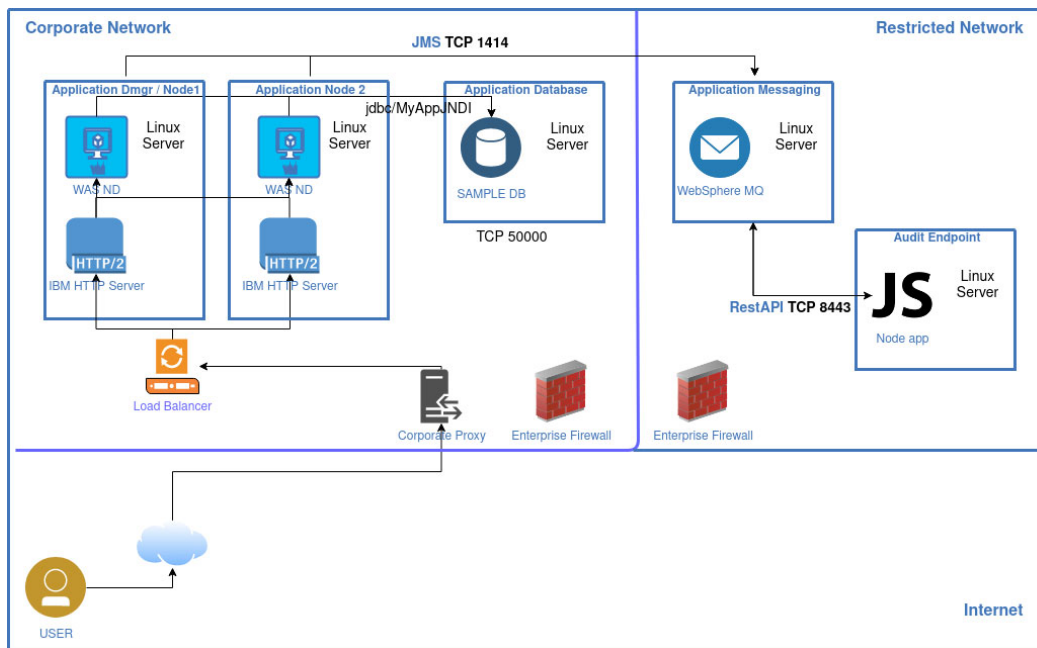


Figure 6-13 An application infrastructure diagram

6.4.2 Why migrate applications

As discussed in Chapter 5, "Migration planning" on page 61, application migration should only be undertaken after thorough planning. There also must be a compelling reason to act, such as the following real world situations:

- ▶ An existing application has outgrown its original platform and is close to reaching the architectural limits of the platform.
- ▶ Software license costs are rapidly increasing as more and more servers are added to an application.
- ▶ Performance issues are arising between distributed application servers and centralized databases.
- ▶ Uncontrolled distributed server growth is leading to power and cooling issues in the data center.
- ▶ Complex distributed systems, which are costly to maintain, are suffering from increasing unreliability.
- ▶ New application development is required following a merger or acquisition.
- ▶ Regulatory requirements impose the need for a more secure environment.

Such situations present valid reasons for considering a migration to a more efficient platform like IBM Z.

In most cases a migration to Linux on IBM Z will help an organization realize significant cost savings over three to five years. The question is, which applications can you migrate and what risk factors are associated with the migration?

The output of this exercise will be a list of an organization's applications ordered by complexity. The list is based on factors such as the number of servers or applications that make up the "IT systems", and can generally be grouped as large, medium, or small applications or number of servers.

6.4.3 Which applications can be migrated

Every computing platform offers specific areas of strength, and the aim of a migration should be to select applications that take advantage of the strengths of the target platform. The classic strengths of IBM Z include high availability, high I/O bandwidth capabilities, the flexibility to run disparate workloads concurrently, and excellent disaster recovery capabilities.

Another key element in choosing the appropriate applications for migration is whether they are supported on Linux on IBM Z. This is normally not a problem with homegrown applications, depending on what language they were written in, but it could be a significant issue with ISV-supplied applications.

6.4.4 Selecting an application for migration to Linux on IBM Z

This section lists and describes the basic rules for selecting an application to migrate to Linux on IBM Z.

The following list includes applications that cannot or should not be migrated to Linux on IBM Z, and explains why they are unsuitable:

- ▶ Applications that are available only on Intel or UNIX platforms.
Requesting ISVs to support their application on IBM Z is a long process. However they can join the [IBM LinuxONE Partner Network](#), which is a program for ISVs to easily port, certify and deploy applications on IBM LinuxONE and Linux on IBM Z platform. ISVs also gain access to Red Hat OpenShift technologies for application development and a rich set of learning resources for skill development.
- ▶ Servers that have already been virtualized.
In such cases, most of the TCO benefits of virtualization have already been realized and only minor benefits will be forthcoming. However, if the existing virtualized environment is reaching its limits or the server leases are within 9 to 12 months of expiry, there may be a good business case for moving the applications to Linux on IBM Z because of its higher virtualization capabilities.
- ▶ Specialized workloads, such as graphics or sound processing.
Graphics and sound processing typically requires specialized hardware and instructions that are not yet available under the platform. Although it is possible to run such type of workloads, the performance might not be beneficial for the user.

The following list includes applications that are suitable for migration and explains why they are suitable:

- ▶ Applications with high I/O or transactional I/O.
Because of its design, IBM Z excels at handling sustained high I/O rates.
- ▶ Applications with lower sustained CPU peaks and average memory needs.

These are ideal workloads for IBM Z. The platform has been designed to run multiple workloads at a consistently high CPU and memory utilization.

- Applications or middleware (database, application servers, and so on) that are supported by a software vendor on multiple platforms, including Linux on IBM Z.

There are no support issues and migration is much simpler.

- Applications that need close proximity to data on IBM Z, or that are components of IBM Z applications.

You can boost the performance and speed of your Linux on IBM Z applications by putting them on the same physical server as their data source.

- Application development environment for Linux on other platforms.

The virtualized Linux on IBM Z platform provides an ideal environment to test applications before their deployment to Linux on other platforms.

6.4.5 Best-suited application for migration

The applications described in this section leverage the IBM Z platform classic strengths, including high availability, high I/O bandwidth capabilities, the flexibility to run disparate workloads concurrently, and excellent disaster recovery characteristics.

Applications that are used to communicate directly with earlier mainframe applications are able to leverage architectural advantages of the IBM Z platform.

IBM software

IBM has ported many of its software products to Linux on IBM Z. The benefit to customers is that a migration from one platform to another is in many cases quite effortless because many of these products share the same code base across multiple platforms. This is particularly the case for IBM WebSphere Application Server, which since Version 6, has had the same code base on Intel x86, IBM POWER, and IBM Z; this simplifies migration considerably.

Linux on IBM Z offers various solutions. For more information, see [Journey to LinuxONE](#).

Generally, migrating from IBM products on distributed servers to the same IBM products on Linux on IBM Z is a relatively straightforward process. You can see examples in Chapter 8, “Hands-on migration” on page 227.

Db2

You can use Db2 for Linux, UNIX, and Windows products on Linux on IBM Z. It works seamlessly in the virtualized environment without any extra configuration. In addition, autonomic features, such as self-tuning memory management and enhanced automatic storage, help the database administrator to maintain and tune the Db2 server. For more information and a migration example from x86, check section 8.2, “Migrating Db2 and its data” on page 229.

A Db2 pureScale environment is aimed at online transaction processing (OLTP) scale-out clusters and can improve the availability and scalability of your database. It is designed to support the strictest requirements for high fault tolerance and can sustain the processing of database requests even under extreme circumstances.

On Linux on IBM Z, Db2 pureScale supports the RoCE network protocol for a high-speed interconnect between members and cluster caching facility (CFs), which, among other IBM Z capabilities, ensures exceptional levels of database availability and scalability. For more information on Db2 pureScale, see [Introduction to a Db2 pureScale environment](#).

Oracle

Because Oracle database is fully supported on Linux on IBM Z and runs in an efficient manner on this platform, it is a good candidate for migration to Linux on IBM Z.

Oracle databases on IBM Z also support Real Application Clusters (RAC), the Oracle high availability (HA) clustering solution. The advantages for Oracle RAC on Linux are a high-availability cluster with low latency within the IBM Z platform that is combined with HiperSockets for inter-LPAR communication.

Oracle WebLogic Server is also supported on IBM Z. It allows you to have a complete Oracle Java environment and high available Oracle database within the same IBM Z.

In many cases, Oracle supports mixed configuration mode where the database tier sits on Linux and applications for Oracle E-Business Suite, Oracle Siebel, and Oracle Business Intelligence run on distributed servers under Linux, Windows, or UNIX. For more information about which Oracle products are certified for IBM Z, contact your Oracle representative or see [Oracle Downloads](#).

Big data Hadoop solutions on IBM Z

Big data and analytics solutions that are built on the IBM Z platform harness the data explosion that is facing businesses. IBM took a leadership role in offering optimized solutions that are ready for immediate deployment. These solutions are built on software, such as IBM BigInsights® and IBM Streams.

IBM also took its leading role in the open source community seriously. IBM made important contributions to projects, such as Apache Hadoop, which enabled continuous development in the fields of analytics and high performance computing. Clients and solution builders that want to innovate on top of a high-performance data analytics platform can take advantage of the flexibility, throughput, and resiliency of IBM Z Platform, and the immediate price-performance value that is provided by IBM Z solutions.

MongoDB solutions on IBM Z

MongoDB's NoSQL technology eliminates the processor burden of object-relational mapping. It enables developers to build and deploy modern applications rapidly, without having to define a data schema in advance and contend with its restrictions. The main features of MongoDB include flexible data modeling, cloud and on-premises cluster management and automation, expressive query language, always-on global deployments, scalability, and high performance.

6.4.6 Other software

This section lists non IBM software that is a good candidate for migrating to Linux on IBM Z.

Infrastructure services

- Network infrastructure, FTP, NFS, DNS, and so on, are very well served on Linux on IBM Z. These workloads are generally minimal, yet they are critical to the business.

The main benefit of hosting these services on Linux on IBM Z is the availability of the hardware's disaster recovery (DR) capabilities.

Additionally, a significant amount of network traffic is generated between data on z/OS and FTP and NFS servers. When the servers are hosted on the same system as the data and HiperSockets is used, then not only is this network traffic greatly reduced, but the batch processing window for that data can also be reduced.

- ▶ LDAP security services fit very well running on Linux on IBM Z, including both OpenLDAP products as well as commercial products like IBM Tivoli® Directory Server, Tivoli Directory Integrator, and Tivoli Access Manager. Using IBM Z architecture, clients can build a robust LDAP infrastructure.

Application development

- ▶ Whether for Java, C/C++, or most other programming languages, a virtualized Linux environment is an ideal platform for application development. Although developers usually develop on a stand-alone platform, testing and modifying are generally performed in a server environment. Developers can be given multiple virtual servers to perform interactive testing while troubleshooting or enhancing the application. z/VM also provides a number of features that enhance application troubleshooting.
- ▶ Other major benefits include the ability to rapidly deploy virtual servers for user acceptance testing and integration testing and, when that is finished, the virtual servers are shut down. If a developer inadvertently “damages” a virtual server, a new server simply has to be cloned. There is no need to spend a great deal of time formatting disks and reinstalling the operating system and required applications.
- ▶ For new applications, virtual servers are deployed quickly and can be easily customized for a specific purpose. Many customers have standard server profiles that are pre-built, so to create a new virtual server, the appropriate profile simply has to be cloned, which can be done in minutes. When an application is discarded for some reason, the virtual servers can be discarded as well.

6.4.7 Selecting an application for a proof of concept

After a business case demonstrates that a Linux on IBM Z migration will provide a positive return on investment (ROI), most clients follow this process:

1. Talk to other customers who have migrated applications to Linux on IBM Z to understand how their migration went and to obtain their recommendations about how to proceed.
2. Choose one of their own applications as a candidate for a proof of concept (POC).

When choosing an application for a POC, it is good practice to keep it as simple as possible because a proof of concept is performed to demonstrate that an application can be successfully migrated to a Linux on IBM Z environment, and that the application results are the same as the production system.

Select an application that is reasonably self-contained and that does not rely too much on inputs from multiple sources and other applications. Also, choose an application that does not require a major rewrite to run on Linux on IBM Z.

The best candidates are applications that are Java based because they are generally platform-independent. However, if you are moving to a different Java Platform, Enterprise Edition specification and a different application server, you may have to make a number of code changes.

Applications written in C/C++ are also suitable if you have the source code because they will have to be recompiled for the IBM Z platform.

After you select an application to migrate, clearly define your goals and expectations. The proof of concept results should achieve the same performance, usability, and functionality as the source production environment.

6.4.8 Applications not supported on Linux on IBM Z

If the application chosen for migration is not supported on Linux on IBM Z by the software vendor, you can ask the vendor for this support. However, it will not happen overnight, so another application might be a better choice for the migration to Linux on IBM Z.

If an unsupported application is required to work with another application that is supported, the best option would be to use a hybrid environment where one application is on Linux on IBM Z and the other application remains on its existing (or modernized) platform and communicates with Linux on IBM Z. For example, suppose that you have a reporting tool that only runs on x86 that analyzes an Oracle database that is also on x86. In this case, the Oracle database could be migrated to Linux on IBM Z and the reporting tool could remain running on x86.

6.4.9 Application interdependencies

Few applications are self-contained. In most cases, an application obtains data from several other applications and its output is sent on to other applications. These applications can also be on different platforms and are often from entities outside your organization. A migration to IBM Z provides an opportunity to simplify your infrastructure without affecting any interdependencies.

Many distributed applications grew in only a few years from a single server to tens or even hundreds of interconnected systems. These interconnected servers not only add network burden, but complexity and built-in fragility. If such an application is being considered for migration, make simplification part of the core of what needs to be done.

Because IBM Z supports all modern communication methods, it is a straightforward process to receive data inputs and transmit data outputs in the same way as before the application was migrated. In this case, no changes to external applications are needed.

Note: The main thing to remember during migration planning is to completely map all application interdependencies. The aim is to identify any obsolete networking technologies and interfaces, which might in turn require another application to be migrated to a current network technology.

6.4.10 Successful application migration

This section outlines the considerations to keep in mind as well as the steps to follow to help you on a successful application migration for Java and C/C++ programs.

6.4.11 Special considerations for migrating a Java application

Migrating Java applications from one platform to another is easy compared to the migration effort that is required for C or C++ applications.

Although Java applications are operating system-independent, the following implementation and distribution specifics must be considered:

- ▶ Most of the Java distributions have their own Java virtual machine (JVM) implementations. Differences exist in the JVM switches. These switches are used to make the JVM and the Java application run as optimally as possible on that platform. Each JVM switch that is used in the source Java environment must be verified for a similar switch in the target Java environment.
- ▶ Although Java SE Developer Kits (JDKs) are expected to conform to common Java specifications, each distribution features slight differences in the helper classes that provide functions to implement specific Java application programming interfaces (APIs). If the application is written to conform to a particular Java distribution, the helper classes that are referenced in the application must be changed to refer to the new Java distribution classes.
- ▶ Special procedures must be followed to obtain the best application migration. One critical point is to update the JVM to the current stable version. The compatibility with earlier versions is significant and performance improvements benefit applications.
- ▶ The Java version lifecycle follows a predictable and well-defined cadence, providing a clear road map for developers and businesses using the Java platform. New major releases occur every six months and are denoted by a three-digit version number like Java 18, 19, 21 etc. These releases introduce new features, APIs, and improvements. Ensure source (x86) and target (IBM Z) use the same and supported Java versions for seamless migration and Opt for LTS (Long-Term Support) releases for extended stability and security.
- ▶ Ensure that the just-in-time (JIT) compiler is enabled.
- ▶ Generally, JVM aligns its page size with the native page size of the OS. On most modern systems, this page size is typically 4096 bytes. This harmony ensures efficient memory management and smooth virtual memory allocation by the JVM, maximizing performance and minimizing overhead. Another advantage of Linux on IBM Z over x86 regarding JVMs is the use of large pages. On Linux on IBM Z you can use 1 MB pages. This reduces the number of pages and therefore saves a large number of CPU cycles searching for this data in memory. To enable large pages, use the `-Xlp` flag along with your JVM as shown in Example 6-1

Example 6-1 Enabling large pages

```
$ java -Xlp -jar mylxapp.jar
```

- ▶ For JVMs with a high memory consumption footprint, set the minimal heap size (`-Xms`) equal to the maximal heap size (`-Xmx`). The size of the heap size should be always less than the total of memory configured to the server.
- ▶ IBM Z runs Java code faster than x86 platforms because of its Pause-less garbage collection functionality, which allows applications to continuously run alongside the garbage collection process.

Note: Starting with IBM Java 8 SR5, enable the Pause-less garbage collection feature by using the `-Xgc:concurrentScavenge` argument to your JVM. For more information about how the Pause-less garbage collection feature works, see the IBM Java SDK documentation.

Understanding the “stop the world” phenomenon

Java's built-in memory management can be a double-edged sword. While it simplifies memory handling, “stop-the-world” garbage collection (GC) pauses can significantly impact application performance and scalability. These pauses halt the entire application, leading to sluggish response times and hindering the application's ability to handle increased workloads.

IBM Z offers industry-leading Java performance through several key features:

- ▶ **Pause-Less Garbage Collection:** this feature eliminates “stop-the-world” pauses during GC, ensuring your applications run faster compared to x86 alternatives.
- ▶ The enhanced cryptographic capabilities of IBM Z further boost performance, allowing you to handle demanding tasks with ease.
- ▶ Unlike x86 systems, IBM Z offers superior vertical scalability, enabling you to handle heavy Java workloads effortlessly by adding more resources to a single server.

6.4.12 Special considerations for migrating C++ applications

When migrating C++ applications, you must be aware of a few special considerations, as explained in this section.

Architecture-dependent code

Programs that are in directories (on non IBM Z systems) with names, such as `/sysdeps` or `/arch` typically contain architecture-dependent code. You must reimplement them for the hardware architecture to port any of these programs to IBM Z.

Assembler code

Any assembler code must be rewritten. Opcodes must be changed to s390 opcodes or, if the code uses assembler header files, you need a suitable version of the header. Linux assembler code for Linux uses the s390 opcodes, but follows the syntax conventions of GNU assembler. The GNU assembler manual is available at [GNU Binutils](#).

ptrace and return structure

Exercise caution when `ptrace` and the return structure are used because they are architecture-dependent.

Little endian to big endian

IBM Z is a big endian system that stores multibyte numbers with the most significant byte at a lower address. Meanwhile, x86 servers are a little endian system, storing the most significant byte at a higher address. Any code that processes byte-oriented data that originated on a little endian system might need some byte-swapping. The data might have to be regenerated or, if that is not possible (for example, shared files), the application might need to be reworked to adjust for processing little endian data.

Changes to build scripts

You must make suitable changes or updates to the `Configuration/build/Makefile` scripts or files, and a requirement to add support for the IBM Z platform.

/proc file system

The proc file system features the following differences:

- ▶ /proc/cpuinfo format is different
- ▶ /proc/interrupts is not implemented
- ▶ /proc/stat does not contain INTR information

Available languages and compilers

Many popular programming languages are available, such as Ruby, Perl, Go, and Python.

6.4.13 Middleware, libraries, and databases

Any middleware or libraries that are needed must be available on Linux on IBM Z. Supported databases include examples of MySQL, Postgres, Oracle, Db2 UDB, and Db2 Connect. As described in 6.4.5, “Best-suited application for migration” on page 98, there is a bunch of middleware available for IBM Z architecture, such as Apache Tomcat, Red Hat JBoss, Oracle WebLogic, and more. For more information about how to install it on the Linux on IBM Z platform, see your product’s documentation.

6.4.14 Helpful steps for an application migration

A successful application migration depends on the combined efforts of the developer team, network team, systems administrators, and any other required technical stakeholders. Without the cooperation of all these groups, it is difficult to achieve a successful migration.

The following overall process might be helpful during your migration:

1. Perform source application mapping.
Start by analyzing the source application, focusing on its suitability to migrate. Consider the following points:
 - Is the source code available to be compiled and deployed on the target server?
 - Is there a version of the middleware available for Linux on IBM Z?
 - Are there performance reports of development tests to compare with after the migration?
2. Design the network solution for the application (see 6.1, “Network analysis” on page 72).
3. Design the file system for the application and middleware (see 6.2, “Storage analysis” on page 84).
4. Clone the Linux server (or servers) from the golden image.
5. Configure the network at the target server (or servers).
6. Create the custom file system at the target server (or servers).
7. Install and configure the middleware at the target server.
8. Copy the application code from the source to the target server.
9. Compile and deploy the application code to the target server.
10. Provide the first application test reports.
11. Start the performance test on the target server to understand the performance of the migrated application.
12. Size the CPU and memory to fit the migration expectations.
13. Run the application stress test.

14. Shut down the source server.
15. Change the IP address and host name of the target server, or change the DNS configuration to the target application server.

6.5 Database analysis

This section provides information about the configurations of the database server on Linux on IBM Z. Best practices for different database software are also presented.

6.5.1 Before database migration

The database server is one of the most highly recommended services to be migrated to Linux on IBM Z. However, it also demands detailed planning because there are technical configuration changes to be considered.

During the migration planning discussions, the workload of the instances and the databases that are running at the source environment must be considered, along with the number of concurrent users and the number of instances and databases running in a unique source server.

6.5.2 Migrating a single instance

For single instance servers, migration is fairly simple because the number of the variables from the source environment to the new destination environment is relatively small. You can use the following steps to migrate when using the same database software vendor and version:

1. Configure the Linux on IBM Z network (follow steps 1 - 4 as listed in 6.1.3, “Helpful steps for a network migration” on page 84).
2. Configure the temporary storage area at the source server and at the destination server.
3. Stop the database services.
4. Issue the export/dump procedures at the source server.
5. Transfer the export/dump files through the network to the destination Linux on an IBM Z server.
6. Shut down the source server.
7. Change the Linux on the IBM Z server host name and IP address.
8. Perform import procedures at the destination server.
9. Perform the database and applications tests.

6.5.3 Migrating multiple instances

For a multiple instance on a single server, or multiple instances on multiple servers, migration is more detailed and complicated. However, among the benefits of the migration are lower license cost, less data center space needed, energy savings, and better performance.

Migrating multiple servers to Linux on IBM Z

A significant factor in the migration of multiple servers to Linux on IBM Z is the distribution of server peak load. Document and compare peak workload information, including how long the

workloads take and how much server resource is used. You can use Table 6-1 to map server workloads when creating the migration configurations.

Table 6-1 Sample database server workload map

Server information			Peak load measure		Peak load time		
Server name	Total of CPU	Total of memory	% CPU used	% Mem. used	Week day	Start time	Stop time

As explained in section 3.6.1, “Virtualized CPU” on page 43, the CPU and memory constraints in an LPAR are possible and desirable, but the server should maintain the same peak load for a long period of time if there are not real CPUs to process each virtual CPU request.

For example, consider a configuration of one LPAR set with three real dedicated CPUs and running three Linux guests. LinuxA has two virtual CPUs, LinuxB has two virtual CPUs, and LinuxC has one virtual CPU.

If LinuxA and LinuxB servers have the same peak load time and period and during this peak load, both LinuxA and LinuxB use 100% of the CPU, that will cause a CPU constraint because the number of virtual CPUs is four and the number of real CPUs is three.

In this case, the z/VM share algorithm will handle all the processor requests and the server still would be available. However, the performance of the application would probably not be very good and would also affect LinuxC’s response time. However, if the server peak loads of LinuxA and LinuxB occur at different times, the entire LPAR will not be affected.

This kind of constraint is acceptable if it happens in intervals of milliseconds to seconds, but it can become problematic in intervals that last for more than a few minutes, depending on how critical the server is to the business purpose.

Having the correct workload capacity plan is key to successfully migrating multiple database servers in a single LPAR on IBM Z.

Another point to consider regarding CPU capacity is the relationship between the source server and the migration server; it is not 1:1. In other words, one distributed server with four CPUs must not necessarily have four CPUs in the destination virtual server; best practice shows that the actual number is less than that. For more information about this topic, refer to 6.5.4, “Technical considerations” on page 107.

Migrating a multiple instance server to Linux on IBM Z

Usually on your development environment you have one database server with multiple instances. This should be all right for a development environment, but when you are migrating your production environment you will want to isolate your instances and simplify the database management. For best results from this type of migration, the workload analysis should be very detailed. Different instances have different workload types, times, and characteristics that might allow the overcommitment of CPUs and memory.

In an environment where the instances are divided among various virtual servers, a software problem occurring on a specific instance will affect only the database server where the

instance is running, so only the database server where the instance is running would need to be restarted or investigated.

It is possible to reduce the number of CPUs allocated to an LPAR by using IFLs. This would result in software licensing savings.

To minimize the work related to database software fixes and security updates, it is possible to use shared devices for database binaries and libraries. For more information about these topics, refer to “Shared file system” on page 93.

Consider the following questions when you migrate from a multiple instance server to multiple Linux on IBM Z virtual servers:

- ▶ Is the source server running at maximal CPU capacity?
- ▶ Is the use of the CPU balanced across all instances? Or is there a unique instance that is consuming all of the CPU?
- ▶ What is the average CPU cycle used by each instance?
- ▶ During which period does the instance use more CPU cycles?
- ▶ Does the instance write or read more data onto the disk devices?
- ▶ How much memory does each instance have allocated?

You can use Table 6-1 on page 106 also to map the instances used by simply changing the Server name column to instance name and documenting the appropriate information.

With this information, you can configure multiple servers in an LPAR to respond to all user requests, without degraded performance and with improved database management. It will be easy to define the number of virtual CPUs that each server needs and avoid the constraint of real CPU in peak usage hours.

Tip: If possible, gather data for an entire month instead for a single day. The more data that you have, the more accurate your analysis will be.

6.5.4 Technical considerations

Database management software requires particularly careful analysis when you are planning a migration. Most database servers use shared memory segments and semaphores to process communications. The database application also uses buffer page configuration to speed up table access and the overall application. In other words, database servers are memory-bound and storage-bound and table access should be considered at server migration.

CPU

The number of virtual CPU resources in a database server is very important; setting the maximum possible does not guarantee better performance. The number of CPUs should be large enough to avoid the processor queue.

The number of processes in a processor queue is influenced by all the other resources of the server, and should not be analyzed as a separate resource. Memory constraints or I/O constraints affect the processor queue number directly, so before deciding that the server does not have enough CPU and adding a new CPU to the service, analyze the CPU schedule time. If the system is running in a high processor queue and most of the CPU time is dedicated to SYSTEM, it probably is associated with memory. The correct parameter to resize

is the memory size. Similarly, if the CPU time is dedicated to I/O WAIT, the file system should be reorganized.

In the beginning, you will not know how many virtual CPUs your database will need on IBM Z. Start with a low number of CPUs and increase as needed.

You can read more about it in *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926.

Memory

The database server uses a very large memory area to achieve acceptable performance, but with Linux on IBM Z, allocating more resources is not related to improving performance. Instead, the machine should be sized as needed and one consideration involved is the server paging process.

Keep in mind that a huge memory setting in the server is not desirable, so at the start of the migration, start the Linux memory size with 60% of the total memory sized from the source server and then increase or decrease as needed.

Swap memory

Specifically in database servers, the swap area should exist and count as part of the total usable memory. However, it should be used only at the peak size to avoid the Linux kernel killing the database process because of memory constraint.

An IBM Z best practice is to use the VDisk devices as swap devices. Because swap configured at VDisk devices provides desirable response time, the eventual memory paging (the process that moves memory blocks to and from real memory and to and from swap memory) is not considered a real problem. It is also not considered a problem if the server has no more than 50% of the swap memory allocated. However, this points to variable paging and swapping allocation, which must be monitored to avoid database outages.

If the server shows a very high paging value for more than 5 minutes, increase memory at the server and continue monitoring the server to find the best memory size.

The Linux server uses the swap memory to allocate memory pages that are not used in real memory as its default configuration. However, that is not the most desirable solution when considering database servers. In fact, it is best to avoid this type of situation. There is a configurable kernel parameter called `swappiness` that determines whether more or fewer pages will be swapped; see Example 6-2.

Example 6-2 /proc/sys/vm/swappiness

at `/etc/sysctl.conf` file include the line
`vm.swappiness = 0`

The configuration displayed in the example will not avoid Linux swapping, but it will reduce the amount of swapping.

The second configuration regarding the swap pages is the `page-cluster` kernel parameters that control the number of pages that will be written at the swap in a single attempt; see Example 6-3. The default value is eight pages at a time. Changing this value to a smaller value will reduce the paging time.

Example 6-3 /proc/sys/vm/page-cluster

at `/etc/sysctl.conf` file include the line

vm.page-cluster = 1

The correct swap size depends on your database and how much memory it uses. The swap memory should only be used in a usage peak, so your swap size should be a safe number that will hold this peak and avoid an outage due out of memory issues. Just for reference, you can use the amount of 20% of the total memory, but do not set more than 2 GB of swap memory at a first moment. Like the memory sizing, you should monitor swap when the usage peak occurs and increase or decrease it accordingly to improve performance.

Shared memory

Linux systems use the interprocessor communication (IPC) facility for efficient communication of process with no kernel intervention. The IPC uses three resources to communicate: messages queues, semaphores, and shared memory.

Shared memory is a memory segment that is shared by more than one process. The size of the shared memory directly influences database performance because if the database can allocate more objects in real memory, the system will perform less I/O.

To obtain the best memory allocation, you must to set some Linux kernel parameters and these parameters depend on what the DBA allocated in the migration. As shown in Table 6-2, some recommendations should be followed to avoid issues like memory starvation.

Table 6-2 Recommended kernel parameters

Parameter	Description	Recommended value
kernel.shmax	Defines the maximum size of one shared memory segment in bytes.	90% of the total memory, but if you have a large amount of storage you can leave 512 MB to 1 GB for the operating system instead.
kernel.shmall	Defines the available memory for shared memory in 4 K pages.	You should convert the shmax value to 4 K (shmax value x 1024 /4)
kernel.shmmni	Defines the maximum number of shared memory segments.	4096. This amount enables large segments to be created avoiding the need for thousands of small shared memory segments. This parameter may vary depending on your application.
kernel.sem	Four values must be set in this parameter. The first one is the number of semaphores, the second indicates the maximum number of semaphores. The third is the maximum number of semaphores operations within one semop call. And the fourth limits the number of allocatable semaphores.	250 256000 32 1024
kernel.msgmni	Maximum number of queues on the system.	1024
kernel.msgmax	Maximum size of a message in bytes.	65536
kernel.msgmnb	Default size of a queue in bytes.	65536

This table is from *DB2 10 for Linux on System z Using z/VM v6.2, Single System Image Clusters and Live Guest Relocation*, SG24-8036, and it was based on the IBM Knowledge Center website:

You can change it to fulfill your database needs.

Storage

Data storage access on a database server is intensive and needs to be considered during server migration. To take advantage of the IBM Z SAP I/O processor, the first consideration in design is to spread the I/O workload over as many paths as possible of the storage server.

In the FICON/ECKD devices, consider using the hyperPAV solution for Linux on IBM Z and a path group with FICON channels. A zFCP solution provides multipath access to the storage device.

Section 3.6.2, “Virtualized disk” on page 44, describes how disk device accesses are made and explains how an external storage system provides its own disk page caching. If such functionality is not used, the Linux OS will spend CPU cycles with disk page caching.

6.5.5 Migrating DB2 and Oracle from x86 to IBM Z

In the following sections, we provide an overview of the steps needed to migrate DB2 and Oracle from x86 to IBM Z. You can find a full example of migrating your DB2 data in Chapter 8, “Hands-on migration” on page 227.

Migrating DB2 databases across platforms

Even though DB2 has many different ways of migrating the data from one operating environment to the target, the simplest and most flexible way of migrating the data is by using the **DB2MOVE** command with the **INSERT** or **LOAD** parameter.

There are four file formats supported for import and export. The format chosen usually reflects the source it comes from or the target tools to be used. Usually the extension of files such as **.ixf**, **.del**, or **.asc** reveal the content format. For example, a file named **employee.ixf** will contain uneditable DB2 UDB interchange format data. Import has the ability to traverse the hierarchy of tables in **.ixf** format.

The following steps present a general overview of how to move an archived database between platforms:

1. Connect to the source DB2 database.
2. Use the export utility to export the database to any of the file formats supported by DB2.
3. Import the exported file to the target environment.

Migrating Oracle databases across platforms

Prior to Oracle 10g, one of the only supported ways to move an Oracle database across platforms was to export the data from the existing database and import it into a new database on the new server.

The following steps present a general overview of how to move a database between platforms:

1. Connect to the source Oracle database.

2. As a DBA user, issue the SQL query shown here to get the exact name of all table spaces. You will need this information later in the process.

```
SELECT tablespace_name FROM dba_tablespaces;
```

3. As a DBA user, perform a full export from the source database, as shown:

```
exp <database name> FULL=y FILE=oradbtst.dmp
```

4. Move the dump file to the target database server. If you use FTP, be sure to copy it in binary format (by entering binary at the FTP prompt) to avoid file corruption.
5. Create a database on the target server. Then, using the DDL Scripts, create the respective tables, indexes, and so on.

Note: Before importing the dump file, you must first create your table spaces, using the information obtained in step 2 of this list.

Otherwise, the import will create the corresponding data files in the same file structure as at the source database, which might not be compatible with the file structure on the target system.

6. As a DBA user, perform a full import with the **IGNORE** parameter enabled:

```
imp <database name> FULL=y IGNORE=y FILE=oradbtst.dmp
```

Using **IGNORE=y** instructs Oracle to ignore any creation errors during the import and permit the import to complete.

This method can require an excessive amount of down time if your database is large. Oracle has developed additional methods to migrate from one hardware platform to another:

- ▶ Transportable tablespaces - introduced in Oracle 8i to allow whole tablespaces to be copied between databases in the time it takes to copy the datafiles.
- ▶ Data Pump export/import - high performance replacements for the original Export and Import utilities.
- ▶ Recover manager (rman) - Oracle Database client that performs backup and recovery tasks on your databases and automates administration of your backup strategies.
- ▶ Oracle GoldenGate - a comprehensive software package for real-time data integration and replication in heterogeneous IT environments.
- ▶ Custom procedural approaches.

6.5.6 Tips for successful migration

Almost all database servers use buffer pools in the shared memory area to manage the database memory context. Avoid using any automatic memory management systems to allocate shared memory. For example, if there is 6 GB of shared memory to be allocated to the database application, force the database application to allocate all memory at the system start.

If the database server is not using all server memory, try to reduce the server memory until the paging process occurs. The first result that indicates insufficient memory size for the Linux servers is swap paging.

If the server for any reason is showing a processor queue, add more virtual CPU to the server. However, monitor the entire LPAR workload to avoid having the performance of a Linux guest interfere with another Linux guest.

The data files and log files must be in different file systems and should be striped across the storage hardware. There should also be multiple paths to the data to ensure availability.

The Linux administrator and database administrator must work together in the Linux guest sizing process because changes may be needed at both the Linux and database levels.

6.6 Backup analysis

This section provides a conceptual approach to migrating backed-up data from an existing operating environment to the target Linux on IBM Z environment.

6.6.1 Introduction to backup and archival concepts

This section gives a high-level introduction to the basic data and storage management paradigms used widely in the IT Industry. It covers data protection or backup, record retention or archiving, storage management, and security.

Backup concepts

The term backup refers to the creation of an additional copy of a data object to be used for operational recovery. As already mentioned, the selection of data objects to be backed up needs to be done carefully to ensure that, when restored, the data is still usable.

A data object can be a file, a part of a file, a directory, or a user-defined data object like a database table. Potentially, you can make several backup versions of the data, each version at a different point in time. These versions are closely tied together and related to the original object as a group of backups. The files are backed up via normal daily backup operations each day that it changes. The most recently backed-up file version is designated the “active” backup. All other versions are “inactive” backups.

If the original data object is corrupted or lost on the client system, restore is the process of recovering typically the most current version of the backed-up data. The number and retention period of backup versions is controlled by backup policy definitions.

Old versions are automatically deleted as new versions are created, either when:

- ▶ The number of versions stored exceeds the defined limit.
- ▶ Or, after a defined period of time.

Common backup types

There are several types of common backups:

- ▶ Normal
- ▶ Incremental
- ▶ Daily

A normal backup copies all selected files and marks each as having been backed up. With normal backups, you need only the most recent copy of the backup file to restore all of the files.

An incremental backup backs up only those files created or changed since the last normal or incremental backup. It marks files as having been backed up. If you use a combination of normal and incremental backups, you need the last normal backup set as well as all the incremental backup sets to restore your data.

A daily backup copies all selected files that have been modified on the day that the daily backup is performed. The backed-up files are not marked as having been backed up.

Archiving concepts

Archiving means creating a copy of a file as a separate object in the storage repository to be retained for a specific period of time. Typically, you would use this function to create an additional copy of data to be saved for historical purposes. For this reason, give special consideration to this task to ensure that the data format is not dependent on anything. Vital records (data that must be kept due to government regulation, compliance, legal, or other business reasons) are likely candidates for the archive process.

The difference between backup and archive software is that backup creates and controls multiple backup versions that are directly attached to the original client file, whereas archive creates an additional stored object that is normally kept for a specific period of time, as in the case of vital records.

6.6.2 KVM backup

Because the KVM hypervisor is a module that is loaded into the Linux kernel, the same tools that are described for a Linux guest in 6.6.4, “Linux backup” on page 114 apply to Linux guests that act as a KVM host. These tools can be used to back up the KVM hypervisor and its system configuration.

In addition, the KVM snapshot and managed save functions can be used to save the state of its managed Linux guests.

6.6.3 z/VM backup

The z/VM hypervisor can use utilities, such as FLASHCOPY and DDR, to back up entire volumes. It is possible to perform a backup of entire Linux volumes when these methods are used. However, because Linux caches its writes in memory before writing changes to disk, it is important to ensure that the guest is shut down or that the volumes to be backed up are mounted read-only at the target server before the backup takes place to ensure that no data is lost.

For more information about options for backing up data within Linux that offer better flexibility, see 6.6.4, “Linux backup” on page 114.

IBM Backup and Restore Manager for z/VM

IBM Backup and Restore Manager for z/VM is a complete solution to back up and restore data for CMS or non-CMS systems (one file, a group of files, or an entire minidisk) in a VM environment. When integrated with the Tape Manager for z/VM, it can compress data during the backup, and support encryption exits.

For more information, see [IBM Backup and Restore Manager for z/VM](#).

6.6.4 Linux backup

Various methods can be used to perform backups with Linux. Whichever method is selected, the output must be a consistent data set that is usable when a recovery is necessary. These methods include command-line tools that are included with every Linux distribution, such as **dd**, **dump**, **cpio**, **rsync**, and **tar**. These tools are useful in the hands of a skilled administrator who has experience using them. The tools have withstood the test of time, but they do require considerable skill to wield effectively.

Other utilities are available that customized the use of the command-line tools. For example, Amanda adds a user-friendly interface for the backup and restore procedures, which makes backup tasks easier to manage. It include a client and server component to facilitate a central backup solution for various remote clients, regardless of the platform. Amanda is typically included, or at least available, in most Linux distributions.

Another useful feature of Linux backups is evident in the capabilities of the file system. File systems, such as ZFS and BTRFS, can take snapshots. These mechanisms can aid the backup process by allowing the backup software to concern itself only with backing up the static snapshot while allowing new changes to the data to continue unimpeded. This process provides for much greater efficiency of the backup process.

Several databases provide mechanisms to create backups, which ensures that memory buffers are flushed to disk and that a consistent data set is created. This feature can also be combined with storage facilities, such as FlashCopy, that perform instantaneous point-in-time copies.

Finally, commercial backup utilities, such as the IBM Spectrum Protect, are available for an enterprise environment. For more information about IBM Spectrum Protect, see [IBM Storage Protect](#).

6.6.5 Migrating backed-up and archived data

When moving to a newer or modern environment, the archived data in the existing environment may no longer be supported, depending on the storage technologies used. It becomes necessary to migrate archived data to a newer format. This ensures compatibility with the production IT environment and maintains data availability.

Why migrate archived data?

Factors that force the migration of archived data include:

- ▶ Preserving data on the same medium would face two problems:
 - The lifetime of the medium.
 - The long-term availability of the technology for reading it.
- ▶ Eventually, the technology change and your solution become less competitive compared to emerging ones.
- ▶ Some older storage technologies have a direct impact on the volume of data that can be stored as well as the space requirements due to the low MBytes/cm³ and Weight/MByte factors.
- ▶ End of support for your current solution.

6.6.6 General archival migration considerations

There are multiple ways of migrating data from the existing operating environment to another operating environment:

- ▶ Change in the hardware environment
- ▶ Change in the hardware and software environment

Change in the hardware environment

This scenario applies when the hardware (servers and storage devices) is replaced by newer and more efficient hardware environments.

Sometimes change in the hardware environment leads to a change of storage technology, which means reorganizing the media data content. Therefore, to allow efficient data retrieval the data inventory structures might need to be reconverted.

Because the operating system and the backup and archival management tools are going to be retained or upgraded, there would not be any incompatibility issues with the archived data. This also means that the migration would be relatively straightforward because the storage backup and archival manager product would be able to access the existing archived data.

Often backup and archival managers have built-in migration tools that will migrate the archived data from the source operating environment to the target new environment. This is a useful point at which to reorganize the archives and purge unwanted data, to efficiently reduce the storage needs of the archives.

Change in the hardware and software environment

This scenario applies when the IT department decides to move to a totally new operating environment (both hardware and software). In this case, both the hardware and software technologies would be replaced. The hardware would have a highly efficient virtualization server and the software would have new technologies that are either proprietary or open source.

This section describes the migration approaches that you can use when the target environment's software stack is changed.

Software is incompatible with Linux on IBM Z

In this approach, because your new guest is not compatible with the old software, all archived data must be restored to a staging server that is compatible with the old backup tool. Use the staging server to restore the archived data and share it with the new IBM Z server that is connected to the new backup software. The following example highlights this process:

1. From the archival software, restore the archived data to a staging server that is compatible with the old backup software.
2. Connect the new server running IBM Z that is used for the current backups and archives to the staging server (for example, by using a shared file system) to access the restored data.
3. The new backup and archival software connects to IBM, accesses the restored data, and rearchives it according to defined organizational attributes and backup policies.

Software is compatible with IBM Z

In this approach, the archived data is restored from the old system to the new IBM Z server. The exported archive data must be rearchived into the new archiving system. You can transfer all the data to the new backup software or transfer it on demand.

6.7 Security analysis

This section discusses the following topics:

- ▶ Security migration overview
- ▶ Code and application analysis
- ▶ Availability and accountability
- ▶ Data integrity, assurance, and confidentiality
- ▶ Security change management
- ▶ Enterprise authentication options
- ▶ CP Assist for Cryptographic Function (CPACF)

6.7.1 Security migration overview

You might assume that simply migrating an application from its existing server to the target Linux on IBM Z server would mean that the security would remain the same. Although that could happen, it probably will not be the case. A major benefit of migrating to z/VM is access to enterprise-class security. Thus, the best time to plan for and take advantage of this benefit is during the migration process.

The security analysis will center around the following areas:

- ▶ Code and application analysis
- ▶ Availability and accountability analysis
- ▶ Data integrity and confidentiality analysis
- ▶ Change and recovery management

Basics of security

Overall security is composed of three domains:

- ▶ Physical security
- ▶ System security
- ▶ Network security

In each domain, the concept of “principle of least privilege” is applied which results in the security policy. That is where each individual is only granted the access that they need, no more. You will need to establish individuals and their roles and who is going to be allowed to do what. This is vital for overall system security because if a compromise occurs, its exposure will only be to the affected role.

Use mandatory access controls to not only ensure that privileged access is given to only what is needed, but to also ensure that authorization is withdrawn when privileges are revoked.

A basic premise underlying the concept of security is that you are only as strong as your weakest point. That is why security is time-consuming, and it is difficult to predict the amount of time that analysis will take. If this is the first time that you are undertaking a security analysis, do not underestimate the time or scope involved in this task.

It is generally held that “security through obscurity” is not a valid method. Using open, well-established security methods implemented correctly provides the best defense. For example, instead of developing your own cryptographic libraries, you should instead use open, established ones that have been vetted for many years. Hiding information creates more system administration work and any mistakes may fail to protect against attacks.

System logs, as well as application logs, need to be immutable. Logs must be kept in such a way that they cannot be altered by system users.

If logs can be altered, overall system integrity will be in question if an impropriety is suspected. Thus it is paramount that all logs be kept in a way that makes them a permanent record of what occurred on the system.

Document the system security and all the assumptions made. Include all “what if” situations that may reasonably be expected to occur. Also, document security plans such as change control, audits, and procedures for break-ins in all domains.

6.7.2 Understanding the z/VM foundation

The Linux virtual machine (VM) is controlled at the z/VM layer. Thus, for a complete security survey to be done, you need both access and an understanding of its security.

The VM layer allows for many Linux images or other operating systems (like z/OS) to run on the same hardware at the same time. The z/VM layer allows for resources to be shared between each VM. It also allows for virtual devices to be created and consumed, like HiperSockets. The highest priority user ID on the z/VM system is MAINT. The MAINT user has root authority and as such must be secured.

IBM Z and existing security policies

Most organizations have an existing security policy dictating that the mainframe must not be Internet-facing. With the migration of a distributed environment to Linux on IBM Z, this often raises questions concerning the role of IBM Z within the existing security policy. A useful approach regarding security policies is to conform with the existing policy as much as possible because it simplifies the migration process. Although usually z/OS is never directly connected to the Internet, this may be a requirement for a distributed environment running on Linux under z/VM in the same IBM Z footprint.

Processor Resource/System Manager (PR/SM) has been certified through the Common Criteria at Evaluation Acceptance Level (EAL) 5+. For more details about Common Criteria, refer to section 1.2.2, “IBM Z strengths” on page 4.

To further ensure the isolation of the z/VM LPAR from the z/OS LPAR, the Open Systems Adapters (OSA) used to connect to external networks by z/VM should be dedicated to the z/VM LPAR. These precautions will ensure that the z/OS environment remains isolated from the Internet. However, if the security policy states that nothing on the mainframe can be connected to the Internet, you have the option of putting the web servers on x86 servers with a physical firewall between the web servers and z/VM.

Firewalls and existing security policies

In many cases, an organization’s existing security policy will identify specific firewalls that have been approved for use on the corporate network. Most often these are hardware firewall appliances. Although z/VM can provide a virtual network between the virtual Linux servers, there is often a requirement to have a firewall between distributed servers, such as an application server talking to a database server. In a distributed environment, the firewall is in the communication path.

For z/VM, there are two options. The first is to implement a software firewall on a virtual server within the virtual Linux environment. This has some challenges because the firewall software may not be used in the organization and as such would have to be certified, which could be a long and complicated process.

The second option is to continue to use the physical firewalls by having the inter-security level communication exit the virtual environment through an Open Systems Adapter (OSA), go through the physical firewall, and then return to the virtual environment via a different OSA. Figure 6-14 illustrates the use of an external firewall.

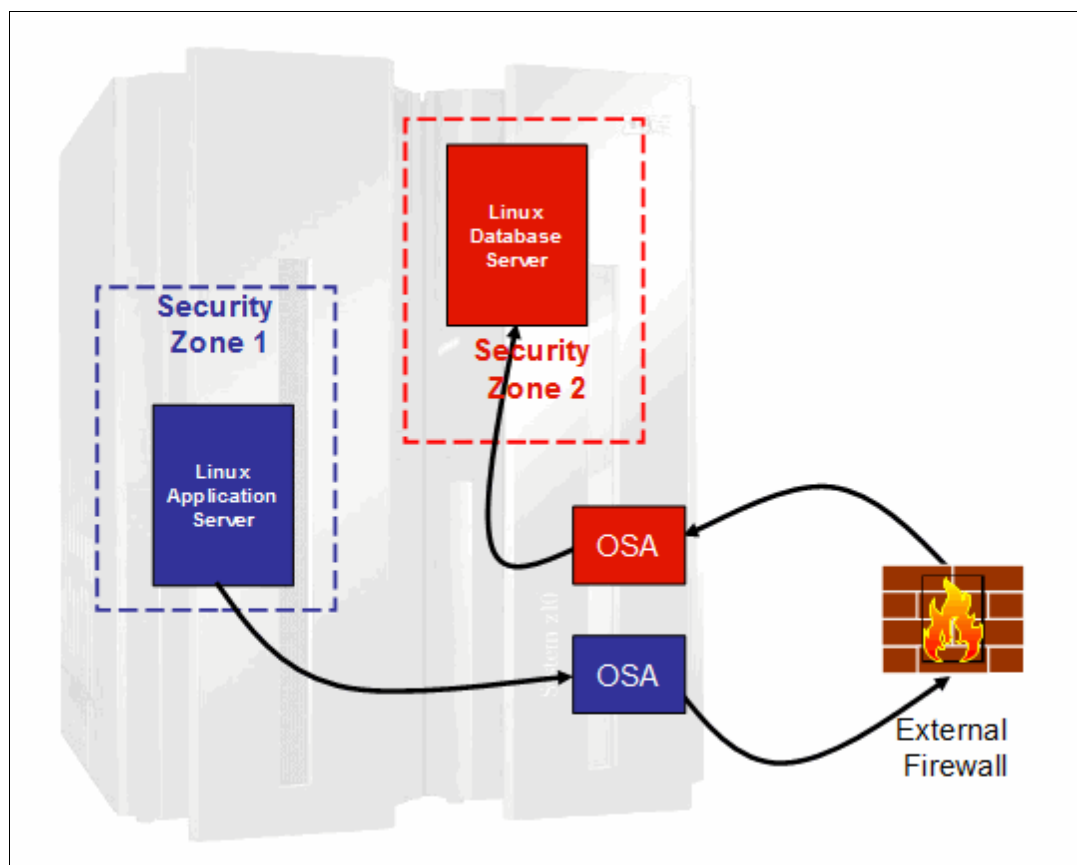


Figure 6-14 Using external firewalls between security zones

In Figure 6-14, the different security zones shown could be in separate LPARs or in the same LPAR. Customers have reported that there is minimal performance impact when using external firewalls.

As mentioned, conforming to the existing security policy can simplify a migration. However, the reality is that for applications within the IBM Z footprint, as shown in Figure 6-14, there may be no requirement for firewalls if all incoming communications to IBM Z are processed by external firewalls.

Control of z/VM

Who will own the z/VM, and what is the protocol for requesting changes or actions? If you will control the z/VM, you need to fully understand z/VM because it is the basis for all the VMs. It must be secure and its access should be highly controlled. Also, a change request protocol should be documented and published to all stakeholders.

You also need to plan for z/VM maintenance, which may require that some or all of the VMs be quiesced. So, ensure that a change window is set aside to allow for maintenance; put a plan in place and a set schedule to allow for security and z/VM updates or maintenance.

Security references

For more information about z/VM and hosting Linux on IBM Z, as well as security and networks, refer to the following IBM Redbooks publications:

- ▶ [Security for Linux on System z: Securing Your Network, TIPS0981](#)
- ▶ [Securing Your Cloud: IBM z/VM Security for IBM z Systems and LinuxONE, SG24-8353](#)
- ▶ [Securing Your Critical Workloads with IBM Hyper Protect Services, SG24-8469](#)
- ▶ [Security for Linux on System z, SG24-7728](#)
- ▶ [Security on z/VM, SG24-7471](#)
- ▶ [Introduction to the New mainframe: z/VM Basics, SG24-7316](#)
- ▶ [IBM Z Connectivity Handbook, SG24-5444](#)

6.7.3 Hardening the base Linux on IBM Z

The term hardening is commonly used in server security to mean the process of taking a generic purpose operating system and changing it to only provide what is necessary for the production environment. This provides a baseline for security for the given operating system.

During migration you may be given an already hardened Linux image, and you will simply need to know what is allowed and not allowed with the image. However, if a hardened Linux image does not exist, you should create and maintain one.

Creating a new hardened Linux on IBM Z

The basics of hardening a Linux on IBM Z consist of removing all unnecessary applications and services, and then securing the applications and services that are left. Explaining this process is beyond the scope of this book, but the following reference may prove helpful to your understanding of this topic: [Security for Linux on System z, SG24-7728](#).

Migrating to a hardened Linux on IBM Z

A hardened Linux on IBM Z should have most if not all applications and services removed or disabled. (Be aware that there may be more than one hardened Linux on IBM Z to choose from, so be sure to choose the version that provides the maximum number of applications and services that you need to perform the migration.)

You will need your migration analysis to determine what needs to be re-enabled. If any applications are to be installed and services enabled, you will need to provide credible business cases for each, individually or as a set. Completing the security analysis can provide just such business cases. Make sure that the documentation includes all applications and services as a delta from the base hardened Linux image.

Important: Red Hat Enterprise Linux includes the SELinux security method, and SUSE Linux Enterprise Server includes AppArmor for its enhanced security method. Determine whether those environments are in use or required, and plan accordingly.

Those mechanisms are very complex, so invest the time to identify code and applications that have not been ported to work in these environments.

Maintaining a hardened Linux on IBM Z

It is necessary to maintain base hardened Linux on IBM Z. Kernels change and security patches are issued, so you need to develop a plan for maintaining the base image and assigning the resources to accomplish it. Thus, successive migrations will benefit from a properly maintained base hardened Linux on IBM Z.

6.7.4 Code and application analysis

Take the time to analyze all code and applications that are being migrated because you will need to know what the current security methods are. You also need to understand what security methods will be used in the target Linux environment, and whether there will be enhancements. Finally, poll the stakeholders to ensure that all migration security requirements will be met.

When moving an application to Linux on IBM Z, consider using as many VMs as you can. That is, separate as much as possible and use the Linux on IBM Z to isolate applications from one another and their data. If many images are available, design the system so that as much separation as possible exists between applications and data. The more isolation, the more straightforward the security will be.

6.7.5 Security issues

This section discusses determining potential security issues when migrating code and applications.

Migrating code

When migrating code, you need to ask whether any known security issues exist. If migrating the code to a Linux on IBM Z that is in an enterprise system, you do not want the application that will be generated from the code to be the weakest link in the system security. All known issues need to be addressed, so plan for it.

Migrating applications

If you know there is a security issue with an application, do not use it. You will need to address all security issues before the system is placed in production. If there are more secure ways to configure an application, invest the time to make those changes during migration; for example, place a database on a different VM than the application using it. Remember, the more separation, the more straightforward security will be. Systems with easy-to-understand security tend to be easier to defend and maintain.

6.7.6 Dependencies

This section discusses determining dependencies before you migrate.

Code dependencies

Almost all code uses APIs and other libraries to carry out the tasks that it was designed for. Thus, you need to review these dependencies before migrating. If you discover that a dependency exists on an item that has a known security issue, you must find and implement a suitable replacement.

Application dependencies

A list of all application dependencies should be generated and reviewed for known security issues. Only fixed or known secure versions should be used. Then, and only then should migration tests be done. Be aware that there will be a temptation to migrate the application over to the new Linux on IBM Z and test to prove that the migration is achievable, but such testing will be invalid if any application or its dependency is on code that has known security issues.

6.7.7 Checking user input

User input is the vector that is most commonly used to attack systems and programs, so all user interaction must be examined carefully. Check all input to make sure that it is within the range of the data needed to be processed. Raw input should never be passed to another application or system request.

Exceptions should also be used. That is, try to ensure that input always conforms to the format that is expected and if the unexpected occurs, that it can be gracefully handled.

6.7.8 Planning for updates when migrating code

When code is migrated to an enterprise-class system, changes need to be addressed in a different manner. Unlike less critical code, changes must be allowed to be executed while the application is still running. Thus, you must ensure that a method is in place to signal that configuration and control files have been updated and need to be reloaded.

There may be a security issue that needs to be addressed by configuration changes. In an enterprise environment, a program should not be stopped but only signaled to take on changes (for example, you might need to change the TCP port that an application uses). Ensure that the code can handle such changes gracefully.

Carefully examine all configuration changes. Do not assume that the changes are valid; verify that they are within the bounds of the setting. If they are not, handle the error gracefully.

6.7.9 Networking

If the code implements TCP sockets, make sure that its design and function are reviewed with the networking team that represents the firewall. That team will probably need to know the following information:

- ▶ What ports will be used by the code, and for what purpose?
- ▶ What type of protocol will be used: TCP, UDP, ICMP, or something else?
- ▶ Will special settings be used on the port, such as TCP keepalive?
- ▶ How long can a connection tolerate a lack of response?
- ▶ How long will a connection be allowed to idle?

6.7.10 Logging and recording events

As previously mentioned, all logs must be kept in a way so that they cannot be changed. They need to be a permanent record of what occurred on the system. Configure the Linux so that syslog (the Linux system log) not only keeps a local record, but also forwards it to a remote secure system. Also, make sure that all critical applications are properly configured to use syslog.

Implementing syslog logging when migrating code

On Linux, syslog-ng will be running. Take time to update the code as needed to send messages to this daemon. At the very least, all information that deals with security should be logged, as well as critical state information. The benefit of implementing syslog functionality is that log maintenance will be performed by the system (as in log rotation and archiving).

6.7.11 Escalations of authority

Apply the “principle of least privilege”; that is, programs should only operate with the authority needed to accomplish a goal. So if the code accesses a database, it should access it only as a user with the access needed, and not as an administrator.

Migrating code

Code should be analyzed to determine where there are escalations of authority. Also, ensure that it accounts for exceptions, so that a de-escalation of authority exists. In other words, make sure that if the code is broken, it does not allow the user to operate at a different access level than is allowed.

Migrating applications

Application programs should not run as root, as this reduces the overall system security. If applications are designed to run with super user privileges, it will be needed to review the accessed resources and remove excessive privileges.

Make sure that server applications are run at the suggested secure settings during all phases of the migration. You do not want to run applications as the administrator while developing, only to discover during testing that certain functions do not work.

6.7.12 Security test plan and peer review

All code and applications that are to be migrated should be in their secure mode during development straight through to test and deployment. It will also be necessary to validate the security assumptions made. This will determine the security test plan. Test everything that can be tested and document what was not tested and why. It is also worthwhile to test change control and verify the restore of backups. If an incident does occur, the only way to recover may be to patch the fault and restore data from the backups (assuming that they have not been compromised).

6.7.13 Availability and accountability

Security involves much more than simply who can access a system. It also involves keeping the system available to authorized users and unavailable to unauthorized uses. Denial-of-service attacks (DoSs) have become more frequent in recent years, and Internet-facing systems must take the possibility of such threats into account.

To implement executable system security there needs to be an audit trail, without exceptions. All access to the system must be logged in a secure fashion to ensure that if an authorized user commits an indiscretion, that it cannot be covered up.

Availability analysis

Sometimes attackers do not break into a system, but instead bring down a service by overwhelming it with requests. Thus system or services availability needs to be understood and service level agreements maintained.

Internet-facing Linux considerations

The Internet is a public “space” where for the most part individuals are anonymous, so every effort must be made to mitigate malicious access if you have an Internet-facing Linux.

As the cost of controlling several distinct IP addresses is getting lower, you may not be able to identify individuals and their IP addresses during an attack, and blocking users based on IP address may not be feasible. It will be needed to work with the networking team to prevent malicious access while still allowing authorized users to have access by employing traffic filtering, anomaly analysis, rate limiting and other approaches. Depending on the service and the surface level, external services may be required.

Communicating availability

Establish a standard for communicating system availability that explains how to report issues and outages to ensure that they are communicated to the appropriate staff. An unexpected interruption in availability can be the first sign that there is a security issue that needs to be addressed.

6.7.14 Accountability analysis

As previously mentioned, all system logs and application logs must be immutable. If attackers gain access, they generally erase evidence of their presence to avoid detection. Also, if users attempt to perform unauthorized acts, they may try to cover their indiscretions by erasing log files or incriminating evidence.

Protecting log files from adulteration

Configure syslog-ng to store logs on a separate secure server. Optimally, the logs should be stored in a Write Once Read Many (WORM) device. Do not delete logs, but keep a secure backup.

Another approach to securing system logs is to use a remote log server, as supported by syslog-ng. See an example of this in 7.7, “Deploying central log server” on page 182. The logs on the remote log server are not necessarily immutable, but they are not directly writeable from a system that has been compromised.

Audit trails encompassing all security domains

Make sure that security audits can be passed at all times by verifying that you can trace an individual’s physical, network, and application access to systems across domains. You must be able to show a system access audit trail from all domains, not just from system access.

Authentication

Ensure that communication end-points are who they say they are. Attackers often “spoof” or pretend to be a system or user that they are not. To protect against such attacks, “authentication” conversations are used:

- ▶ Users must be assured that they are connecting to the server they think they are.
- ▶ Servers need to be assured that users are who they say they are.
- ▶ This authentication must be kept private so that eavesdropping cannot occur.
- ▶ Avoid the use of self signed certificates, as they can easily be forged.

For remote SSH access, consider using Certificate Based Secure Shell Authentication when possible. Unlike keys, certificates are tied to an user and have an expiration date. They contain metadata, allowing for role-based access control, and can disable certain SSH options (like port forwarding).

TLS is widely known for securing HTTP, but it can be used to protect other protocols as well. Internal services should implement TLS if supported, such as LDAP traffic, databases, FTP and SMTP. If a protocol does not support TLS encapsulation, consider employing an encrypted tunneling solution (such as **stunnel**) to keep data secure in transit.

All certificates used on TLS and SSH must be issued by an internal CA, and all systems must be configured to trust the internal CA. Users should not expect to connect to a system and receive an “invalid certificate” error message, and have to choose to trust the certificate anyway. This behavior allows attackers to issue look-alike certificates and trick users into trusting a fake certificate.

6.7.15 Data integrity and confidentiality

A benefit of migrating to Linux on IBM Z is that data can be stored on an enterprise-class system. However, you need to analyze the current state of the data and then determine how it will fit in the new enterprise system.

Data integrity analysis

Data integrity refers to the assurance that data is unchanged from creation to reception. Data integrity also entails understanding the following items:

- ▶ Who can access what data and what is allowed
- ▶ Whether there is an audit trail in place to map who changed what and when
- ▶ Whether the data is corrupted in some way and how is it to be restored
- ▶ Whether there is a disaster recovery plan in place

Protecting data at rest from unauthorized access

The most used method on Linux for data at rest encryption is called Linux Unified Key Setup (LUKS). LUKS ensures the data is always encrypted on the storage devices, and can only be read using the correct key. It protects the data even if the media is stolen or compromised.

IBM Z Crypto Express cryptographic co-processors securely generate, store and manipulate cryptographic keys. It ensures that even the operational system cannot access the plain values of the cryptographic keys, and all cryptographic operations are executed on a secure module inside the IBM Z CPU. For more information, refer to [Pervasive Encryption for Data Volumes, SC34-2782-04](#).

You should also prevent offline copies of a database from being transferred to unauthorized devices or systems. The use of Data Loss Prevention tools can help detect if sensitive data is being copied by accident or malice. Linux Audit Daemon (**auditd**) can generate log entries on events happening on the system and help detect security violations.

Data backups: part of security

Part of your security plan needs to include backups and how they are stored. They need to be kept in a secure way. When backups are kept separate from the system for disaster recovery purposes, use encryption to prevent unauthorized access. Understand the impact if the backups are stolen and mitigate the risk.

Creating backups is only part of the solution. You need to be able to restore backups in a timely manner, and choose the correct backup approach. Disk-based backups are easy to implement but may create corrupted files on the image. Incremental backups are faster to create and use less space, but take more time to recover. Full backups are faster to recover, but use more space and take more time.

Design the backup policy to meet both the Recovery Time Objective (RTO) and the Recovery Point Objective (RPO). RTO is the acceptable amount of time for recovering a system after a disaster, and RPO is the acceptable amount of data loss caused by a disaster.

6.7.16 Confidentiality analysis

Confidentiality must first be communicated and then enforced. Thus, before users can access a system they need to be told what the confidentiality of a system is and how any data or information will be used or shared. Then, a system needs to be in place to enforce the policy. This is normally done by auditing access logs. If a violation is detected, it will need to be communicated to the affected parties.

Understanding laws and regulations before an incident occurs

Before you can create a confidentiality policy, you need to understand what is legally expected:

- ▶ Are there national, regional, or state laws that need to be followed?
- ▶ Are there any industry compliance requirements (such as Payments Card Industry (PCI) requirements) regarding the storage of credit card information?
- ▶ Is there a company policy? If so, it needs to be followed.
- ▶ Document all expectations regarding how long to keep the data (for example, “We expect or are required to keep the data for up to 5 years.”).

Publishing your confidentiality policy

You need to communicate the confidentiality policy in such a way as to ensure that all users of the system are aware of it and thus can be held accountable. When a user logs in to a system, use the Message Of The Day (MOTD) found in `/etc/motd` as shown in Example 6-4 to communicate with your system users.

Example 6-4 Use /etc/motd to communicate system policy

```
*****
*      .--.      Welcome to the Linux s/390x VM      *
*      |o_o|      SUSE Linux Enterprise Server 11 SP3*
*      |:~/|      System Admin: John Doe             *
*      // \ \      jdoe@company.com                  *
*      (| |)      This system governed by corporate  *
*      /'\_/_\    Policy K49-r v21 please read       *
*      \__)=(__/  before accessing system           *
*****
```

Tip: Use ANSI art or special characters to make the login window attractive. It is useful to display system information such as the Linux distribution with its version and release information, along with a greeting.

On web pages, create a link from the main page so that the system policy can be easily accessed. If you are allowing VNC login, display the policy by updating `/etc/gdm/custom.conf` as shown in Example 6-5.

Example 6-5 Policy found in /etc/gdm/custom.conf

```
[greeter]
DefaultRemoteWelcome=false
RemoteWelcome=Connected to %n must read policy K49-R v21
```

Having a plan in place before an incident occurs

Have a plan in place in case confidentiality is violated. The plan should include:

- ▶ What is considered a security incident.
- ▶ Who should be notified and what should be disclosed about the incident.
- ▶ If there is a requirement to notify the public, document how and what should be disclosed.

Communicate actions that will be taken to prevent future incidents.

6.7.17 Security change management

No system is perfect so there will be changes, however infrequent. Because security fixes are important to keep current, there should be a plan to understand their impact on the system. If a Linux needs to be restarted, it must be done in an orderly and timely basis.

After the system is moved from test to production mode, it will remain that way. Outages are expensive for companies, but failing to plan change windows and downtime will also cause security problems. In the rare case that a VM needs to be restarted, you need the ability to allow for these types of changes.

Testing changes with a clone of the Linux on IBM Z

The advantage of migrating to a Linux on IBM Z is that you can clone a VM and test changes before applying them to the production images. Run through the complete change from start to finish, rather than assuming it will work.

Record how long it takes to make changes and test worse case scenarios (also keeping track of the time). After testing the change on the clone is complete, you will be able to report to production stakeholders how long the change will take and how long the worst case will take.

6.7.18 Enterprise authentication options

Migrating to an enterprise system means that user and identification management can be consolidated. In this section, we describe enterprise authentication options and where to find the corresponding information explaining how to implement them.

A common centralized LDAP server

When migrating applications and code to a Linux on IBM Z, you can simplify user administration by storing user information in a Lightweight Directory Access Protocol (LDAP) server. Configuring the Linux on IBM Z to authenticate from a centralized LDAP server provides the following benefits:

- ▶ User management is simplified; users can be managed across the enterprise.
- ▶ A centralized LDAP server helps on audit trails and logging.
- ▶ Changes made to a user will be applied across all images.
- ▶ An offline VM could contain outdated user information. Using LDAP assures that bringing an old image online will not compromise current security.

LDAP server on z/OS means RACF integration

If IBM RACF® is used to manage user information, then installing LDAP on a z/OS system will allow LDAP access to RACF. In turn, this allows a single, highly secure repository of user information in RACF and lets that information be exposed to Linux VMs via an LDAP server. For more information, refer to [Security for Linux on System z, SG24-7728](#).

You can also configure Samba to use LDAP as its user repository. Thus, you can have one security domain across MS Windows, IBM AIX® and Linux, with IBM Z as the core. For more information about this topic, refer to *Open Your Windows with Samba on Linux*, REDP-3780.

6.7.19 Integrated Cryptographic Service Facility

When migrating to Linux on IBM Z, the underlying hardware has the ability to accelerate cryptographic operations. The CP Assist for Cryptographic Function (CPACF) supports synchronous cryptographic functions. The work is processed by the crypto-assist processor that is integrated into every processing unit (PU) of every IBM Z or the Crypto Express card, if it is installed.

The supported APIs are listed here.

OpenCryptoki

An open source implementation of Public-Key Cryptography Standard #11 (PKCS#11), OpenCryptoki uses the libica shared library to access IBM cryptographic adapters through the z90crypt device driver.

OpenSSL

An open source implementation of Secure Sockets Layer, OpenSSL can utilize the libica shared library for hardware encryption.

Global Security Kit

Provided as part of the IBM HTTP Server, Global Security Kit (GSKit) manages SSL certificates. It utilizes OpenCryptoki for hardware encryption.

Using this approach will offload the cycles and allow for more concurrent access to a web server that is using SSL or applications that use one of supported APIs. Refer to *The Virtualization Cookbook for IBM Z Volume 2: Red Hat Enterprise Linux 8.2*, SG24-8303, to learn how to configure your system so that your Linux on IBM Z will take advantage of the installed hardware.

6.8 Operational analysis

The source application will usually come with a complete support structure. Depending upon the application, this support could be 24 hours a day, 7 days a week, 365 days a year. The application will rely upon manual and automated intervention to start, stop, monitor, and maintain the services provided by the application. It may also be a requirement of receiving vendor support that regular operational tasks are performed, such as archiving log files.

This section describes some of the operational issues which, if present in the source application, must be addressed in the target application. A careful and detailed analysis about how the source application is supported by operations staff is required for a successful migration effort.

An analysis of the operational functions may highlight characteristics of the application that were not clear from the analysis of other application interfaces or from the code itself. The application code may be successfully ported, but it is just as important that the application's operational support structures be migrated successfully as well.

6.8.1 The operational environment

Operational environments present many tasks and challenges to the operations staff, who are often required to multi-task when monitoring consoles and managing other physical equipment. For this reason, it is important to ensure that the migrated application fits in smoothly with the current operational environment.

Operational tasks might be affected by the source application migrating to the target application running on Linux on IBM Z.

6.8.2 Operational migration tasks

This section describes operational issues that might change when migrating the source application to the target application in a new environment:

- ▶ Starting and stopping the application

These processes can be automated or manual. The source application probably had certain methods for starting and stopping its processes, but the target application will probably have different commands and methods for starting and stopping the application.

If the target application is a manual process, the operators must be trained and the appropriate documentation must be written and published. If it is an automated process, the automation scripts need to be written, tested, documented, and explained to the operators (including guidance on what to do if the automation does not operate).

- ▶ Notification of problems

Sometimes operators can receive automated messages or indicators that they are unfamiliar with and do not know how to respond to. Operators need to know who to turn to for guidance when this type of problem arises, so the application owner needs to be clearly identified. If the application owner is unavailable or unresponsive, escalation procedures need to be in place. These details might change when the application is migrated to the target system.

- ▶ Normal intervention and monitoring

Some applications need to be checked or modified during their lifecycle throughout the day. Often this simply involves monitoring indicators or displays that show the health of the application. New procedures for the migrated target application must be communicated to the operators. Hands-on training sessions are optimal for operators as they learn by observation and perform required tasks.

- ▶ Hardware manipulation

Some migrations will include hardware consolidation or replacement. Operators will need to be trained on how to operate and manipulate the new hardware. Even if the operators are not required to manipulate the hardware, it is still useful to let them know what is running on the new server and to have the appropriate documentation, labels, and signs available for reference.

- ▶ Hardware intervention and problem escalation

There are fewer hardware interventions for operators to deal with on IBM Z.

For example, with the source application and server, an operator might be comfortable with and even required to reboot a server by using the power switch. On IBM Z, however, it is a serious error to use a power switch to react to a server or application problem.

If there is a new hardware vendor in the migration project, the method that the operators must use to notify the vendor of an actionable message or event needs to be communicated to the operators. A test of that procedure should be carried out and then

documented. You should not wait for a critical situation to occur before learning how to contact vendors or other support personnel. The contact information should include day shift, off hours, and weekend names and numbers. The requirements for the vendor contact should be clear. The vendor often requires precise, detailed information such as serial numbers, machine type, location.

- ▶ **Batch procedures and scheduling**

Most applications will have batch processes that support the application. Automatic scheduling software is common at most installations to schedule and track those batch processes. Schedulers within the operations department will be involved to create the necessary scheduling changes for the migrated application. The new schedules will then be communicated to the operators on each shift.

- ▶ **Other considerations**

Not everything in your operating environment can be envisioned and described here. The intent of this chapter is to give you an idea of possible operational issues related to the migration project. Think of everything in your operating environment that may change or be affected by the migration of the source application to the target application. Then, create a plan to perform the requisite operational migration tasks. And finally, execute your plan.

6.8.3 Single system image and live guest relocation

As seen in 3.4, “Single system image and live guest relocation” on page 39, single system image (SSI) and live guest relocation (LGR) will simplify z/VM systems management. From the operational side, it will enable software or hardware maintenance and upgrades without disruption to the business. Operators or z/VM system administrators are able to move guests to other members that are on the same or separate IBM Z servers. No disruption is necessary.

Tip: The LGR can also be used for workload balancing.

6.8.4 z/VM and virtual machine management products

A number of operational tasks will have to be performed in the migrated environment, such as:

- ▶ Display and manage virtual servers and resources.
- ▶ Provision Linux guests, network, and storage.
- ▶ Capture and clone virtual servers across LPARs and CPCs.
- ▶ Activate and deactivate z/VM guests in the current z/VM system.
- ▶ Lock or unlock z/VM resources.
- ▶ Create and configure virtual switches (vSwitches) and guest LANs.
- ▶ Provide storage management and provisioning for z/VM and Linux.
- ▶ Run shell scripts or REXX EXECs directly from the user interface for more customized management and provisioning.
- ▶ Support advanced z/VM capabilities such as SSI and LGR. Perform a live guest relocation of one or more z/VM guests.

A number of products are available, from IBM and others, to provide some of these capabilities. Some, such as IBM Directory Maintenance Facility (DirMaint) are foundational and support other functions, while some stand and provide their own function. Some examples of products available are:

- ▶ IBM Directory Maintenance Facility for z/VM (directory manager)
- ▶ IBM Performance Toolkit for z/VM (performance analysis)
- ▶ IBM Resource Access Control Facility for z/VM (external security manager)
- ▶ IBM Operations Manager for z/VM (console management and automation)
- ▶ Broadcom CA VM-Secure (directory and external security manager)
- ▶ Velocity Software zPRO (web-based z/VM and virtual machine management)
- ▶ IBM Cloud Infrastructure Center (IaaS manager based on OpenStack)
- ▶ Log-On Wave (GUI-based z/VM and virtual machine management)

The Open mainframe Project also hosts projects that can assist in managing workloads on z/VM:

- ▶ Feilong (REST API capability for z/VM functions)
- ▶ Tessia (scripting and automation of virtual machine actions on z/VM)

6.9 Disaster recovery and availability analysis

IT system outages can significantly impact businesses by rendering critical systems unavailable. The key to ensuring that this problem does not occur is to analyze your systems and determine a hierarchy of availability needs. Keep in mind that not everything needs a remote hot site.

For better understanding, the following terms and definitions are used when discussing disaster recovery, high availability, and related concepts:

- ▶ DR

Planning for and using redundant hardware, software, networks, facilities, and so on, to recover the IT systems of a data center or the major components of an IT facility if they become unavailable for some reason.

- ▶ HA

Provide service during defined periods, at acceptable or agreed upon levels, and mask unplanned outages from users. High availability employs fault tolerance, automated failure detection, recovery, bypass reconsideration, testing, problem, and change management.

- ▶ Continuous operations (CO)

Continuously operate and mask planned outages from users. Continuous operations employs nondisruptive hardware and software changes, nondisruptive configuration changes, and software coexistence.

- ▶ Continuous availability (CA)

Deliver nondisruptive service to users 7 days a week, 24 hours a day. With continuous availability, there are no planned or unplanned outages.

The goal for mission-critical systems should be continuous availability. Otherwise, the systems should not be defined as mission-critical.

6.9.1 Availability analysis

Migrating an application to a virtualized Linux environment on IBM Z offers an opportunity to implement an availability profile in line with the impact of the unavailability that the application has on the organization's overall business. Sometimes, however, such an analysis is not straightforward. For example, test and development workloads are generally not considered to be mission-critical. However, because they may be needed to correct an error in a production system, consider providing for some sort of test and development environment in your DR planning.

The challenge with DR is to achieve a balance between the impact of an unavailable system on the health of the business versus the cost of creating a resilient environment for the application. This planning should include the likely scenarios that could impact an application's availability, as well as unrelated events that could impact the ability of a business to function.

The usual IT issues such as server failure, network failure, power outage, disk failure, application failure, and operator error, can be planned for through duplication of resources and sites. Unrelated factors are rare and not directly related to IT, but they can have a huge impact on the ability of a business to function. These events include fire, natural disasters such as earthquake, severe weather, and flood, as well as civil disturbances, which can have a major impact on the ability of people to go to work.

Although this chapter focuses on the IT-related issues, you should also have a plan in place to deal with other, non-IT related events.

6.9.2 Single points of failure

In determining the DR requirements of an application, you need to look at the probability of failure of a component as well as the cost to eliminate a single point of failure (SPOF).

Table 6-3 lists the components of an IBM Z virtualized environment running an application under z/VM and Linux and the relative costs of rectifying a single point of failure.

Table 6-3 Potential single points of failure that can impact availability

Single point of failure	Probability of failure	Cost to rectify
IBM Z hardware	Very low	High
IBM Z LPAR	Very low	Low
z/VM	Low	Low
Linux	Low	Very low
Disk system microcode	Low	Medium
Virtual network within z/VM system	Very low	Low
Physical network	Medium	Medium
Application	High	Very Low

Apart from hardware and software failures, there are other outages that can impact an application's availability. These planned outages are:

- Hardware upgrades requiring a power-on reset

- ▶ LPAR configuration changes requiring a reboot of the LPAR
- ▶ z/VM maintenance
- ▶ Linux kernel maintenance requiring a reboot
- ▶ Application maintenance

6.9.3 IBM Z features for high availability

IBM Z has been designed around providing HA. Perhaps the most design effort has gone in to the transparent recovery of processor errors. In the event of a hard processor error at an individual core level, the task is moved to a spare processor where processing continues transparently to the application. In the IBM z16, a number of availability features have been introduced to reduce the number of planned system outages. For example, the following actions are now fully concurrent and require no system outage:

- ▶ Adding LPARs
- ▶ Adding logical processors to a partition
- ▶ Adding logical channel sets (LCSSs) - I/O paths
- ▶ Adding subchannel sets
- ▶ Enabling dynamic I/O
- ▶ Adding a cryptographic processor to an LPAR

Additionally, many services enhancements have been introduced to avoid planned outages:

- ▶ Concurrent firmware fixes
- ▶ Concurrent driver upgrades
- ▶ Concurrent parts replacement
- ▶ Concurrent hardware upgrades

The IBM z16 offers a number of customer-initiated capacity on demand features. These billable features are designed to provide customers with additional capacity to handle the following events:

- ▶ Customer-Initiated Upgrade (CIU) is used for a permanent capacity upgrade.
- ▶ Capacity BackUp (CBU) is predefined capacity for DR. A system at a DR site does not need to have the same capacity as the primary site. In the event of a declared disaster, or for up to 5 DR tests, the customer can turn on the number of processors, including IFLs, required to handle the workload from the primary site.
- ▶ Capacity for a Planned Event (CPE) is used to replace capacity lost within the enterprise due to a planned event such as a facility upgrade or system relocation.

On/Off Capacity on Demand provides extra capacity in 2-hour increments that is available to be turned on to satisfy peak demand in workloads.

Note: For more information about the IBM z16, see [IBM z16 \(3931\) Technical Guide](#), SG24-8951.

6.9.4 Availability scenarios

The following scenarios present a number of different situations where a Linux on IBM Z environment is set up with increasing degrees of availability and increasing levels of cost. The key to maximum availability is to eliminate single points of failure.

In all scenarios, it is assumed the IBM zEnterprise® System is configured with redundant LPARs, redundant channel paths to disk (FICON and FCP), redundant Open System Adapters connected to the organization’s network, redundant system consoles, and redundant Hardware Management Consoles. This is the normal setup for an IBM zEnterprise System.

The application design should include redundant software servers. The storage infrastructure should also include redundant FICON directors, redundant Fibre Channel switches, mirrored disks, and data.

The communications network should be designed around redundancy with redundant network routers, switches, hubs, and wireless access points.

Remember that for mission-critical systems, an uninterrupted power supply should also be provided as well as a second site far enough away from the primary site to avoid being affected by natural disasters.

Another important factor in the availability of applications is security and access controls. For more information about this topic, refer to 6.7, “Security analysis” on page 116.

Single IBM Z LPAR: Clustered WebSphere Application Server

Figure 6-15 shows an IBM Z LPAR sharing system resources to all Linux virtual machines in the LPAR. The WebSphere Application Servers are in a two-node cluster. If the Integrated Facility for Linux (IFL) fails, IBM zEnterprise System will automatically switch the workloads to a spare or any unassigned processor without any disruption to the active task.

If a Linux virtual machine running the WebSphere Application Server workload fails, the other node in the cluster will take over if you are running WebSphere Application Server Network Deployment. This is achieved because an application deployed to a cluster runs on all members concurrently. Additional availability is provided through the nondisruptive addition of new VMs to the cluster.

Note: z/OS is optional in the first six scenarios.

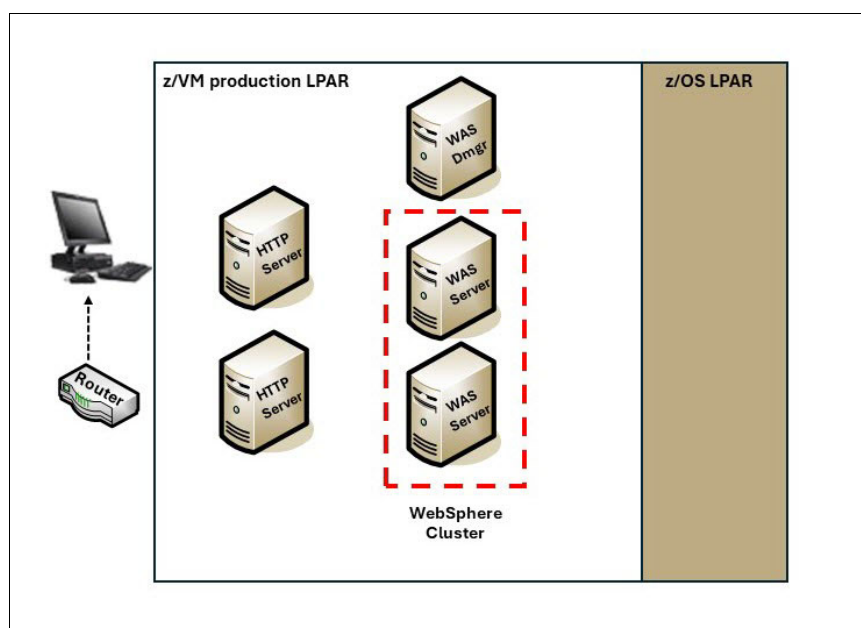


Figure 6-15 Single LPAR WebSphere Application Server cluster

This environment also provides additional availability through redundant HTTP servers.

Multiple LPARs: HA solution for Linux on IBM Z

Figure 6-16 shows a scenario where there are three LPARs defined. Each LPAR could have a dedicated IFL or a single IFL, or multiple IFLs could be shared among all LPARs. The LPAR weight determines the relative priority of an LPAR against other LPARs.

In this case, the production workload and WebSphere Application Server cluster is split across two LPARs, which give HA to WebSphere Application Server because an LPAR or z/VM failure will not impact the availability of WebSphere Application Server.

Development and test workloads run in their own LPAR so any errant servers will have no impact on the production workloads. As in the first scenario, a failure of a IBM Z IFL will be rectified automatically without any impact to the running application.

This configuration eliminates most failure points at a reasonable cost.

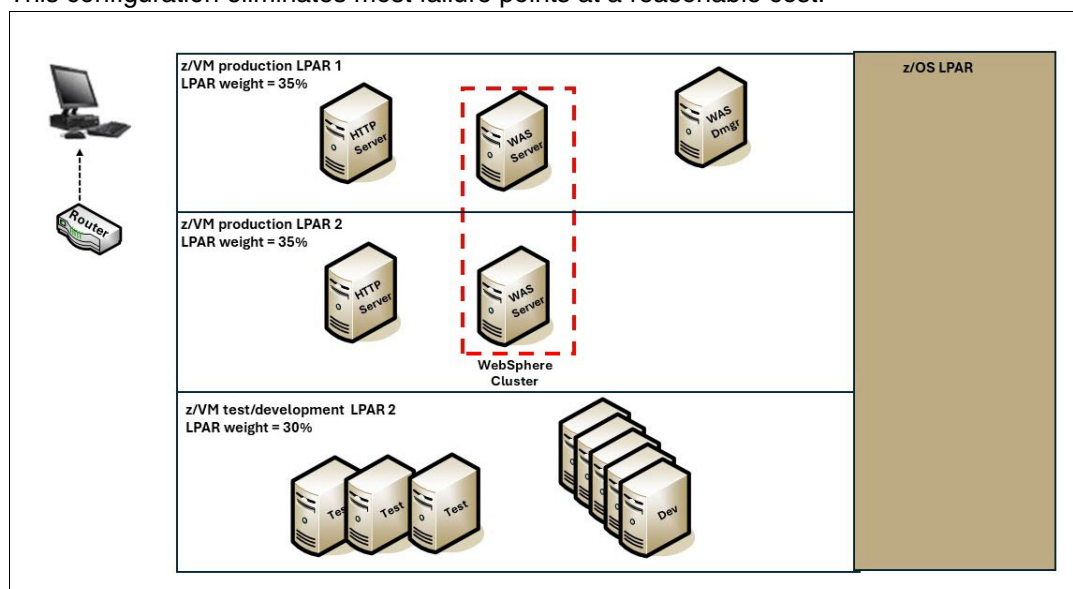


Figure 6-16 HA Linux environment on IBM Z

Active/cold standby cluster

Figure 6-17 on page 135 describes another approach in which, instead of having redundant virtual servers, an active/cold standby cluster is established. In this case, Tivoli System Automation for Multiplatforms (SA MP) monitors the servers and in the event of an outage will automate failover to the cold standby server.

SA MP runs on each node in the cluster. It monitors cluster nodes and exchanges information through Reliable Scalable Cluster Technology (RSCT) services. SA MP also creates a Service IP address as an alias on an appropriate network adapter on Node 1 where the HTTP server will be started.

Only one instance of the HTTP Server is defined to SA MP to be able to run on either of the two nodes with a “depends on” relationship to a single IP address (the Service IP). SA MP starts the HTTP Server on Node 1 and at user-defined intervals invokes a script to confirm that it is still up and serving pages. It also monitors the Linux node itself to ensure it remains active.

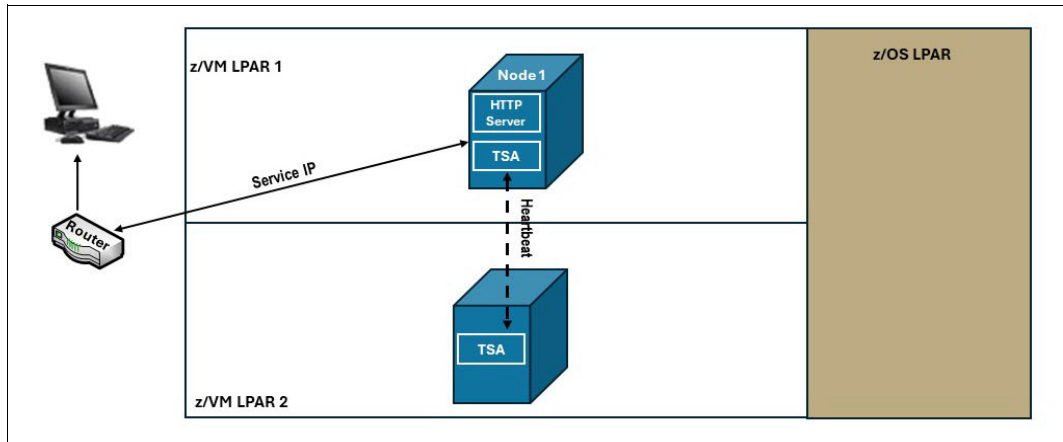


Figure 6-17 Normal situation: Tivoli System Automation monitors for outages

When a failure occurs, RSCT determines that Node 1 is no longer responding. SA MP then moves the Service IP over to Node 2 and restarts the HTTP server there, as illustrated in Figure 6-18.

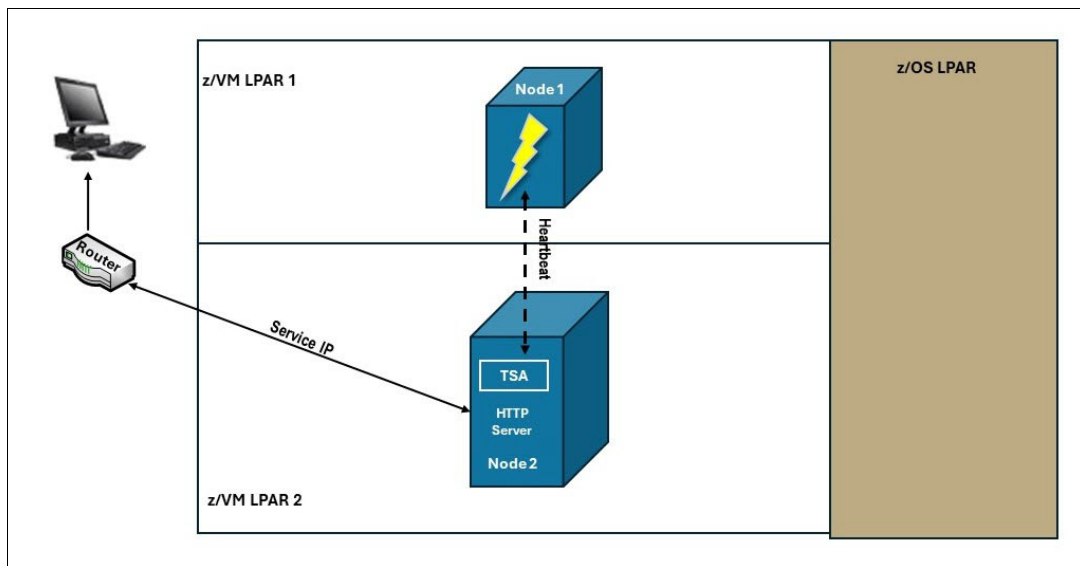


Figure 6-18 Outage occurs: Tivoli System Automation fails over to cold standby

Active/active application server cluster

Figure 6-19 on page 136 shows the WebSphere Application Server setup in an active/active configuration where the WebSphere Application Server Cluster spans two Linux virtual machines in two LPARs. This setup handles the very rare occurrence of the failure of an LPAR. More importantly, it also allows z/VM maintenance to be performed without an outage to the WebSphere applications. In this case, the Linux servers and z/VM are shut down in LPAR 2. An initial program load (IPL) is done of z/VM with new maintenance applied and the Linux virtual machines are restarted and the WebSphere cluster is restored. This task would be scheduled for a time when the processing load is light. Live guest relocation could also be used to avoid an outage due to z/VM maintenance.

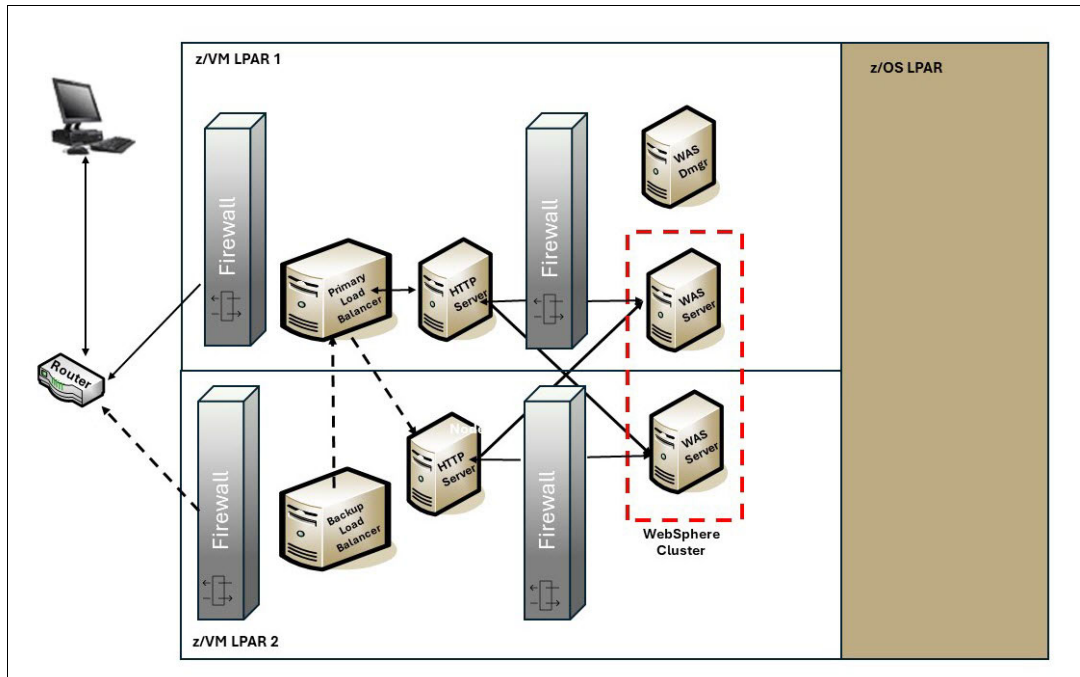


Figure 6-19 Active/active WebSphere Application Server cluster

Active/active WebSphere Application Server cluster with database replication

Figure 6-20 on page 137 shows a Db2 database added to the active/active WebSphere cluster. To provide high availability for the Db2 database, the Db2 data replication feature, High Availability Disaster Recovery (HADR) is used. HADR protects against data failure by replication changes from the source database (called primary) to a target database (called standby).

In the event of a z/VM or LPAR outage of the primary Db2 system, the standby Db2 system will take over in seconds, thus providing high availability. Communication between the Db2 primary and Db2 standby systems is via TCP/IP, which in this case would be done using the IBM Z high speed virtual network feature HiperSockets.

The standby Db2 system can also be at a remote site to provide enhanced availability in the event of a site failure.

IBM Tivoli System Automation for Multiplatforms (SA MP) running in both Db2 servers is designed to automatically detect a failure of the primary, and it issues commands on the standby for its Db2 to become the primary.

Other cluster management software could be used. However, SA MP and sample automation scripts are included with Db2 to only manage the HA requirements of your Db2 database system.

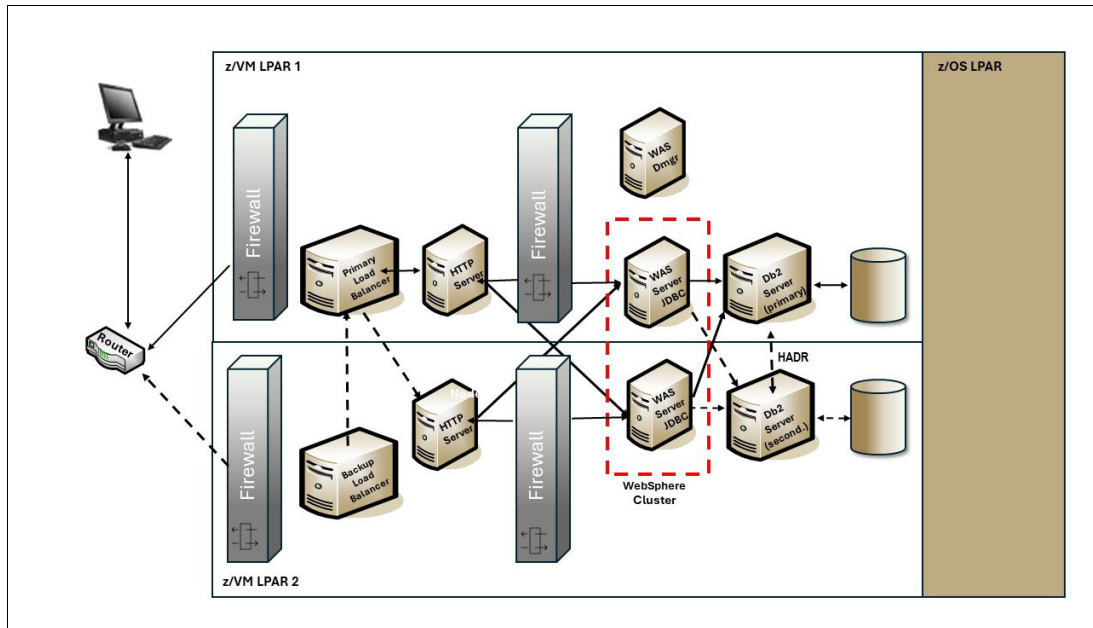


Figure 6-20 Active/active WebSphere Application Server cluster and Db2 HADR

Active/active WebSphere Application Server cluster with database sharing

Figure 6-21 on page 138 shows that database sharing was introduced using Oracle Real Application Clusters (RAC). Oracle RAC provides HA for applications by having multiple RAC nodes sharing a single copy of the data. If a cluster node fails, the in-flight transaction is lost but the other server in the RAC can receive all Java Database Connectivity (JDBC) requests.

In a IBM Z environment, communication between the database nodes would use a virtual LAN in the same LPAR or HiperSockets to other LPARs. Both methods are at memory-to-memory speeds with very low latency.

For more information, see [Oracle RAC](#).

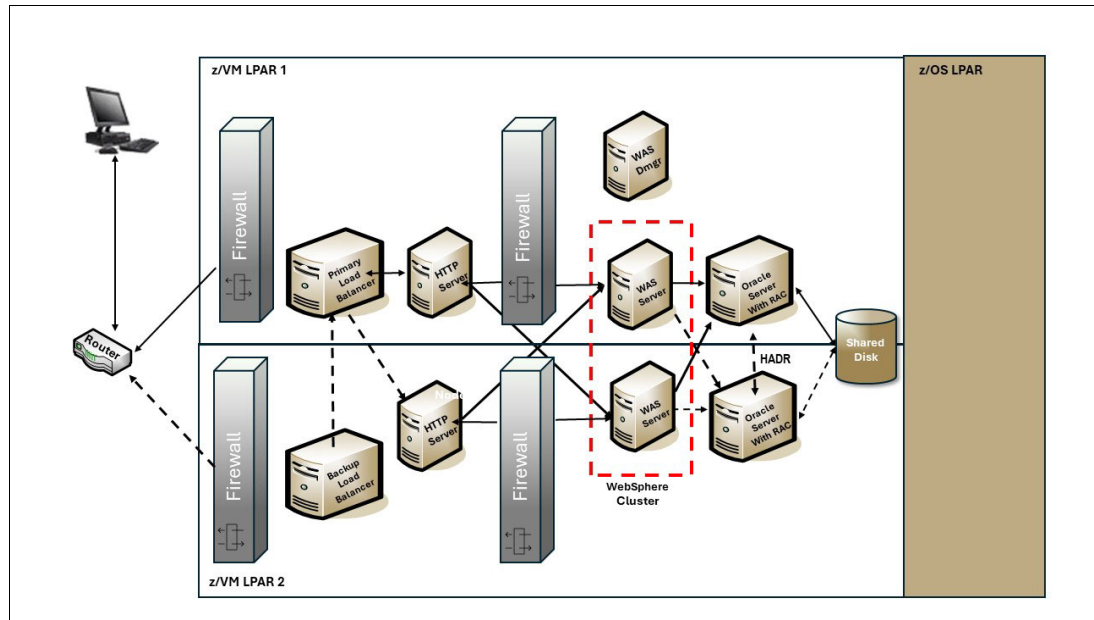


Figure 6-21 Database sharing using Oracle RAC

Active/active WebSphere Application Server cluster with Db2 sharing in z/OS Parallel Sysplex

In Figure 6-22 on page 139, we introduce the additional benefits provided by the z/OS IBM Parallel Sysplex®. Briefly, a Parallel Sysplex is an HA configuration designed to provide CA of systems and applications. In the case of Db2 data sharing, the Parallel Sysplex allows all members of the sysplex update access to shared data by using a centralized arbitrator known as the coupling facility (CF).

Each WebSphere Application Server is configured to use the JDBC Type 4 driver for communication with the Db2 z/OS data sharing members. It is sysplex-aware and works cooperatively with Db2 and the z/OS Workload Manager (WLM) on z/OS to balance workloads across the available members of the Db2 data sharing groups.

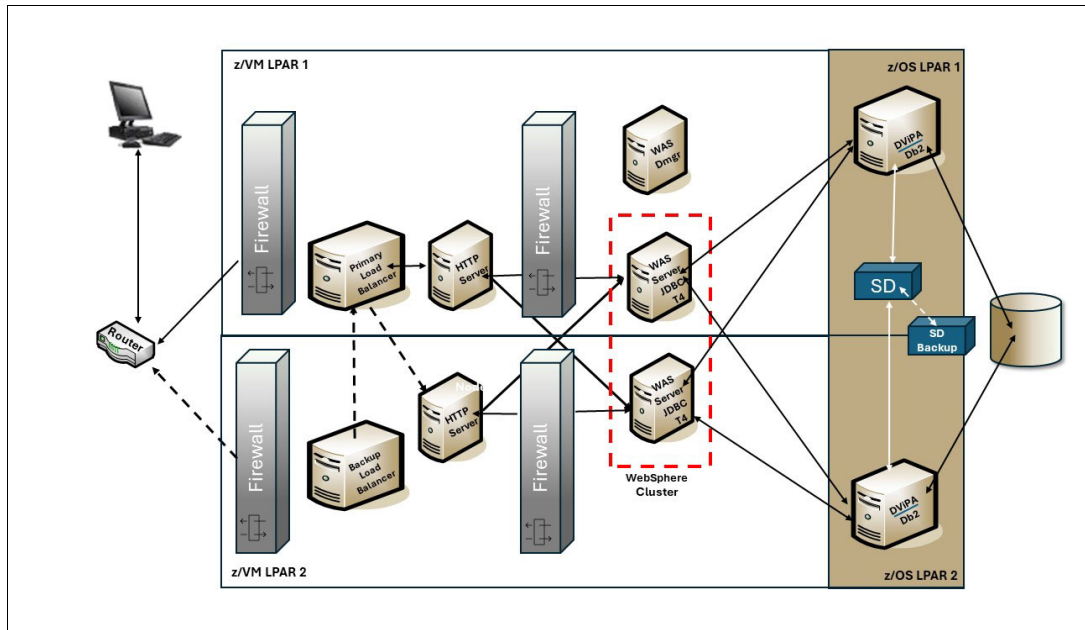


Figure 6-22 Database sharing using z/OS Parallel Sysplex

Note: SD: Sysplex Director. Provides an initial contact single cluster IP address (known as Dynamic VIPA or DVIPA in Figure 6-22) for the data sharing group. After initial contact, all subsequent communication is directly between the JDBC T4 client and the Db2 data sharing group members.

Active/active WebSphere Application Server cluster with database sharing on z/OS across cities

For the ultimate availability solution, it is possible to have two sites up to 100 km (62 miles) apart and provide full Db2 data sharing between WebSphere Application Server clusters at each site. The key element in this solution is Globally Dispersed Parallel Sysplex (IBM GDPS) Metro Mirror. GDPS Metro Mirror uses a feature on the IBM ESS800 and IBM DS6000 and DS8000® family of storage systems called Peer-to-Peer Remote Copy (PPRC).

All critical data resides on the storage subsystem (or subsystems) in Site 1 (the primary copy of data) and is mirrored to Site 2 (the secondary copy of data) via Synchronous PPRC. With Synchronous PPRC, the write to the primary copy is not complete until it has been replicated to the secondary copy. PPRC is designed to make it possible for a site switch with no data loss.

The primary Controlling System (K1) running in Site 2 performs the following services:

- ▶ It monitors the Parallel Sysplex cluster, Coupling Facilities, and storage subsystems, and maintains GDPS status.
- ▶ It manages a controlled site switch for outages of z/OS and Parallel Sysplex, z/VM, and Linux on IBM Z (as a guest under z/VM).
- ▶ It invokes IBM HyperSwap®² on z/OS and z/VM for a site switch of disk subsystems, which can eliminate the need for an IPL at the recovery site to use the mirrored disks.

² HyperSwap is a z/OS feature that provides for the continuous availability of storage devices by transparently switching all primary PPRC disk subsystems with the secondary PPRC disk subsystems for planned and unplanned outages.

- It works with Tivoli System Automation Multiplatform across z/VM and Linux to understand their state and coordinate their restart during the recovery phase.
- It invokes network switching, based on user-defined automation scripts.

Figure 6-23 shows that Site A and Site B are in a GDPS and share the same Db2 data. GDPS helps to automate recovery procedures for planned and unplanned outages to provide near-Continuous Availability and Disaster Recovery capability.

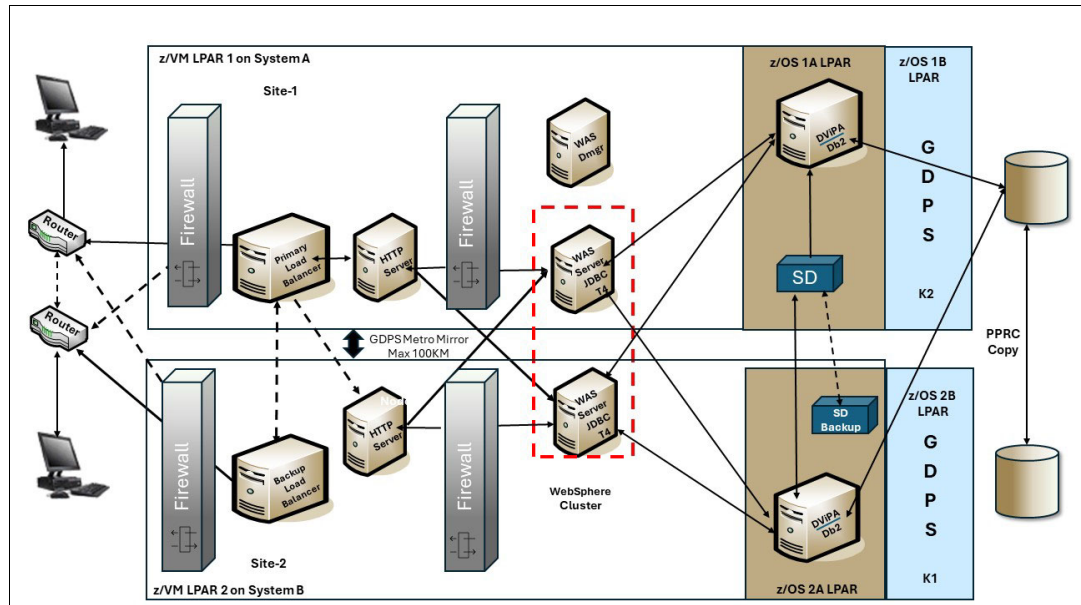


Figure 6-23 GDPS solution for near-continuous availability

Distances greater than 100 km (62 miles) require an asynchronous copy where the application resumes after the write is performed to the primary copy. The write to the remote mirror takes place later, so it is not synchronized with the primary copy. More detailed description of this topic is beyond the scope of this book.

6.9.5 Linux-HA Project

The Linux-HA Project provides HA solutions for Linux through an open development community. The majority of Linux-HA software is licensed under the Free Software Foundation GNU Public License (GPL) and the Free Software Foundation GNU Lesser General Public License (LGPL).

For more information, see [Linux-HA](#).

Note: For more details about Linux-HA and examples of its use in a z/VM Linux on IBM Z environment, refer to [Achieving High Availability on Linux for System z with Linux-HA Release 2, SG24-7711](#).

6.9.6 HA add-ons provided by SUSE and Red Hat

There are two solutions that are worth your attention when you are looking for a High Availability third-party solution.

Depending on the distribution that you have chosen for your environment (SUSE or Red Hat), you can find add-ons that will facilitate the HA implementation.

- ▶ [SUSE Linux Enterprise HA Extension](#)
- ▶ [Red Hat Enterprise Linux High Availability Add-On](#)

6.9.7 Understanding the availability requirements of your applications

This section describes how service-level agreements (SLAs) and the cost of providing availability can help you achieve a better understanding of the availability requirements of your applications.

6.9.8 Service level agreements

To determine the availability requirements of applications that you want to migrate to Linux on IBM Z, you must take into account the needs of the business units that rely on these applications. Ideally, SLAs are in place that state requirements, such as availability needs, response time, maximum system utilization, DR requirements. This should be the basis for the design of the target system on Linux.

If SLAs do not exist, before starting to design a solution, discuss with the business units what levels of service you can offer and what level of investment they are willing to make. The key to the success for an SLA is that it is both achievable and measurable with defined penalties for failure to deliver. You also need to ensure that there are regular reviews because things will change.

According to IT Service Management principles, an SLA would typically define or cover the following topics:

- ▶ The services to be delivered
- ▶ Performance, tracking, and reporting mechanisms
- ▶ Problem and change management procedures
- ▶ Dispute resolution procedures
- ▶ The recipient's duties and responsibilities
- ▶ Security
- ▶ Legislative compliance
- ▶ Intellectual property and confidential information issues
- ▶ Agreement termination

Some of these components might not be relevant in an "in-house" SLA.

From an availability view point, an SLA for an "in-house" business application should focus on the first two items, name what service is being delivered and how is it being measured:

- ▶ Application availability hours, for example:
 - 24 hours/day x 7 days a week
 - 6:00 am to 6:00 pm, weekdays
 - 9:00 am to 5:00 pm, weekdays, and so on
 - Definition of how availability is measured and who will do the measurement. For example, system availability, application availability, database availability, network availability
- ▶ Minimum system response time
 - Defined number and definition of where and how is it measured

6.9.9 The cost of availability

As shown from the examples in this chapter, there is a great degree of difference in cost and complexity of the various availability options discussed. Providing CA and a DR plan is not an insignificant expense but with the degree of reliance on IT systems by most businesses today, it is a cost that cannot be ignored.

If you have a web-facing revenue-generating application, you can calculate the cost of downtime by simply monitoring the average revenue generated over a period of time. This provides an idea of the amount of revenue that may be lost during an outage and how much you should spend to make the application more resilient. Other businesses will have different ways of calculating the cost of downtime.

Keep in mind that for any HA configuration to be successful in a real DR situation, there needs to be a fully documented DR plan in place that is fully tested at least once every year.

6.10 Linux on IBM Z cloud management

The following applications are available to help you with IBM Z cloud management:

- ▶ OpenStack
- ▶ vRealize Automation for Linux on IBM Z and Linux on IBM Z
- ▶ IBM Cloud Infrastructure Center

OpenStack

To provide cloud management capability, both z/VM and KVM are OpenStack-enabled, which is the industry standard for ubiquitous cloud computing platforms. Applications that use the OpenStack APIs are supported on both hypervisors. IBM Cloud Infrastructure Center (CIC) is an infrastructure as a service software product that helps establish a cloud infrastructure and simplifies management of a virtualized environment. It also offers additional services such as identity management and orchestration accessed in the same programmatic manner through the API.

vRealize Automation for Linux on IBM Z

With VMware vRealize Automation (vRA), you can extend your existing vRealize Automation cloud to IBM Z. vRealize Automation interfaces with the OpenStack enabled cloud endpoints from IBM Z to provide cloud management services. This configuration validates the openness of IBM Z to allow clients to use the same cloud management that is used within their distributed cloud environment

IBM Cloud Infrastructure Center

[IBM Cloud Infrastructure Center](#) is an infrastructure management software designed to provide on-premises cloud deployments for enterprises. It's specifically tailored for Linux on IBM Z and LinuxONE platforms, offering a range of functionalities, such as:

- ▶ Efficiently manage the lifecycle of virtual machines on Linux on IBM Z / IBM LinuxONE, inclusive of network and storage resources.
- ▶ Comprehensive capabilities encompass multi-tenancy, high availability (HA) clustering, backup and restore functionalities, live migration, consistency grouping, snapshots, and more.
- ▶ Leverage industry-standard OpenStack APIs for seamless and flexible automation and operations, enabling Infrastructure as Code practices.

- ▶ Seamlessly integrate with various cloud platforms and tooling, such as user service portals, IBM Instana™, IBM Cloud Pak® for Watson AIOps, Terraform, VMware vRealize, and others.
- ▶ Simplify the deployment and management of Red Hat OpenShift clusters, either as part of IBM Cloud Paks or as standalone Red Hat OpenShift instances, using Ansible Playbooks.

IBM Cloud Infrastructure Center provide the layer to manage the IBM Z/IBM LinuxONE based Infrastructure-as-a-Service (IaaS).

The Figure 6-24 shows the intersection between the infrastructure and the cloud platform, where Cloud Infrastructure Center is acting.

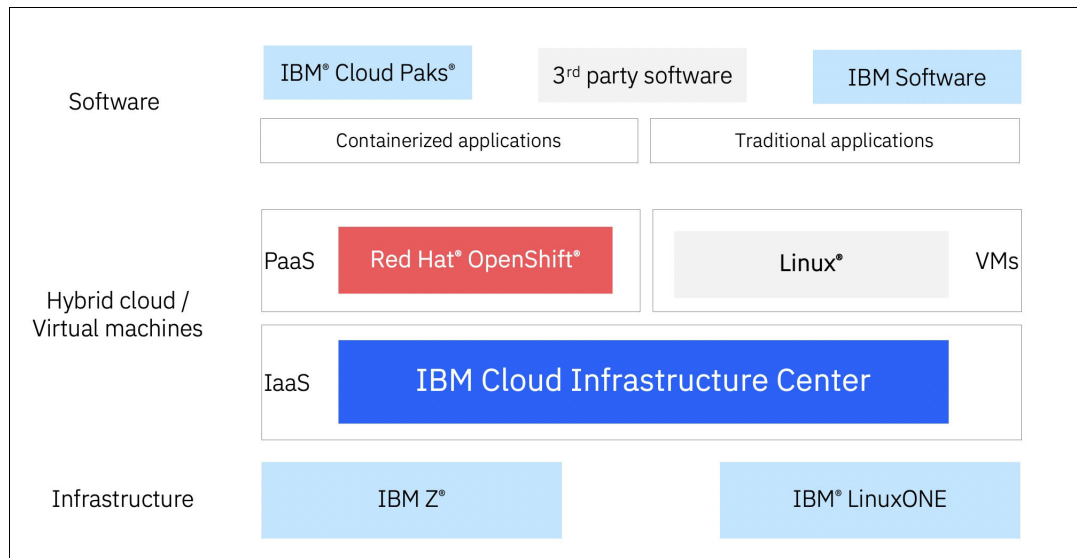


Figure 6-24 IBM Cloud Infrastructure Center

For more information, see [IBM Cloud Infrastructure Center 1.2.1](#).

For a sample configuration of z/VM and KVM installation see [Hybrid cloud with on-premises cloud on IBM Z or LinuxONE](#), SG24-8530.



Deployment of workloads

In this chapter, we provide examples of deploying workloads using mainframe (HA) clustering as well as deploying the application MediaWiki and MySQL. We provide an example of deploying OpenLDAP, a central log server, and a file and print service.

There are many things to analyze and consider leading up to the deployment of workloads to the mainframe. When the proper planning is completed, the migration should move smoothly.

As mentioned in section 6.3, “VMware to KVM migration options” on page 94, there are many workloads that represent a “perfect fit” on IBM Z. Not all can be demonstrated in this book. The migration of some very practical applications, such as IBM Db2, is illustrated as a hands-on exercise in Chapter 8, “Hands-on migration” on page 227. Mission-critical applications, ERP, CRM, business intelligence, and more, are clearly what you want to run on IBM Z, but only generic examples can be included in a guide such as this. Following the guides, the checklists, and the information contained previously in this book, and using this chapter of examples, will help lead you to success.

Standard infrastructure applications are also very well suited on IBM Z, and these are just as critical. In this chapter, the deployment of some standard services is demonstrated. Such an illustration of deploying standard services should likewise represent a pattern that can be followed.

The following main sections are available in this chapter:

- ▶ 7.1, “Deciding between containers and VMs” on page 146
- ▶ 7.2, “Deploying HA clustering” on page 148
- ▶ 7.3, “Setting up Docker” on page 148
- ▶ 7.4, “Deploying MongoDB on Linux on IBM Z” on page 156
- ▶ 7.5, “Deploying MediaWiki and MySQL” on page 165
- ▶ 7.6, “Deploying OpenLDAP” on page 175
- ▶ 7.7, “Deploying central log server” on page 182
- ▶ 7.8, “Deploying Samba” on page 206
- ▶ 7.9, “Deploying Terraform” on page 209
- ▶ 7.10, “Deploying Apache Kafka” on page 211
- ▶ 7.11, “Deploying Validated Open Source Software” on page 216
- ▶ 7.12, “Containerized Workloads” on page 217

7.1 Deciding between containers and VMs

Linux on IBM Z is equipped with fast general-purpose processors, ideally suited for data processing throughput. The large number of cores available in Linux on IBM Z and their high input/output bandwidth means that open-source solutions can scale up and scale out.

Although the underlying hardware is ready for a highly scalable environment, advantages and disadvantages exist that are specific to having the solution on a container or virtual machine (VM). Containers can allow you to have many more applications in a single physical server than a VM can. However, a business might need application deployments that are based on VMs. All aspects of the enterprise application must be considered before deciding whether to run it under containers or in a single VM.

The following are the deciding factors for determining whether the solution should be on containers or VMs:

- ▶ **Application packaging:** If you want to run multiple copies of a single app such as MongoDB, use containers. However, if you want the flexibility of running multiple applications (MongoDB with a Java based homegrown application), use a VM.
- ▶ **Dependencies:** Usually, containers tend to lock in to a particular version of an operating system and its subsystems and libraries. This feature can be an advantage for an administrator, because with containers you can create a portable, consistent operating environment including programs and libraries for development, testing, and deployment. From a VM perspective, no matter what hypervisor you use, you can deploy any operating environment. This feature is especially useful with in-house applications with specific dependencies.
- ▶ **Resources:** From a resource perspective, containers share an operating system, kernel instance, network connection, and base file system. Each instance of the application runs within a separate user space. This configuration significantly cuts back on the CPU usage that is associated with running multiple operating systems because a new kernel is not needed for each user session. This is one of the major reasons why containers are often used for running specific applications.
- ▶ **Automation:** Concerning speed to production, with the advent of the cloud and DevOps mode of application development, containers have an advantage because each container provides a microservice and can be part of a larger solution. This feature provides containers with the advantage of scale over the VM.
- ▶ **Security:** Without any alterations to the container, a VM is more secure than a container. VMs have the advantage of featuring hardware isolation, whereas containers share kernel resources and application libraries. This feature means that if a VM breaks down, it is less likely to affect other VMs in the same operating environment. For now, regular containers do not have hardware isolation. If your organization has high security requirements, stick with VMs.

Table 7-1 shows some aspects to consider when deciding between containers and VMs.

Table 7-1 Containers vs. VMs

Aspect	Containers	Virtual Machines (VMs)
Isolation	Lightweight, OS-level isolation.	Heavyweight, hardware-level isolation
Resource Usage	Shares host OS kernel, less overhead	Requires separate OS, more resource overhead

Aspect	Containers	Virtual Machines (VMs)
Performance	Generally faster startup and execution	Slower startup and execution due to overhead
Portability ^a	Easily portable between environments	Less portable, tied to specific hardware/OS
Flexibility	Can run multiple containers on host	Each VM runs separate OS, less flexible
Resource Sharing	Shares host resources efficiently.	Resources allocated to each VM independently
Security	Potential for container escape exploits.	Stronger isolation, less susceptible to attacks
Scaling	Easier to scale horizontally	Scalability limited by VM infrastructure

a. It is crucial to construct multi-architecture containers to ensure portability across diverse computing environments (such as x64,x86, s390, ppc64, arm64 ppc64le,...).

These examples illustrate how containers and virtual machines cater to different types of workloads based on factors such as application architecture, resource requirements, security considerations, and compatibility needs:

► Containers

- Web Servers: Containers are ideal for deploying web server applications like Apache HTTP Server or Nginx. Each container can encapsulate the web server along with its dependencies, making it easy to deploy and scale.
- Microservices: Containerization is well suited for microservices architectures where applications are broken down into smaller, decoupled services. Each microservice can run in its own container, enabling easier development, deployment, and scaling.
- CI/CD Pipelines: Continuous Integration and Continuous Deployment (CI/CD) pipelines often use containerized build environments. Containers provide consistency across different stages of the pipeline and allow developers to package their application along with its dependencies.
- Dev/Test Environments: Containers are valuable for creating lightweight, isolated development and testing environments. Developers can quickly spin up containers with the necessary dependencies to test their code, leading to faster development cycles.
- Stateful Applications: While traditionally considered more challenging for containers, modern container orchestrators like Red Hat OpenShift support stateful workloads. Stateful applications like databases or key-value stores can be containerized for easier management and scalability.

► Virtual Machines:

- Legacy Applications: Virtual machines (VMs) are suitable for running legacy applications that may have complex dependencies or compatibility requirements. VMs provide a more traditional environment similar to physical hardware, allowing legacy applications to run without modification.
- Resource-Intensive Workloads: Workloads that require dedicated access to hardware resources, such as high-performance computing (HPC) applications may benefit from running in virtual machines. VMs can be provisioned with specific CPU, memory, and storage resources to meet the workload's demands.

- **Security-Sensitive Workloads:** Virtual machines provide stronger isolation between workloads compared to containers. Security-sensitive applications or workloads that require strict isolation boundaries can be deployed in separate VMs to minimize the risk of security breaches or data leaks.
- **Workloads Requiring Different Operating Systems:** Virtual machines support running multiple operating systems concurrently on the same physical hardware. Workloads that require different operating systems or versions can be deployed in separate VMs, providing flexibility in the choice of operating environment.

In real-world scenarios, it's common to have workloads where certain components are better suited to run in containers, while others are best deployed in virtual machines (VMs).

Consider a web application that consists of a front-end (for example, React.js) built by using modern microservices architecture (for example, Spring Boot) and a legacy backend database (for example, Oracle database). In this scenario, the front end microservices can be containerized and deployed using container orchestration platforms like Kubernetes. Containerization offers agility, scalability, and ease of deployment for the front end components.

On the other hand, the legacy backend database, which requires strict data consistency and stability, may be better suited to run in a VM. Hosting the database in a VM ensures strong isolation, dedicated access to resources, and compatibility with legacy software dependencies.

Most organization run a mix of both containers and VMs in their clouds and data centers. The economy of containers at scale makes good financial sense. At the same time, VMs still have their virtues and use cases. Linux on IBM Z provides the best in class features for running containers and VMs.

7.2 Deploying HA clustering

Both Red Hat and SUSE deliver add-on products that provide mainframe failover clustering for their respective Linux operating systems. These products can be deployed to increase the availability and reliability of Linux workloads on IBM Z. In the event of a catastrophic event that takes one cluster member down, the applications running can be automatically brought online on one of the partner members of the cluster, with no perceived downtime. SUSE Enterprise Linux mainframe Extensions (HAE) even offers geodistribution clustering, enabling the restart of a database or application from a remote site.

7.3 Setting up Docker

The Docker engine is available for installation on SUSE and Ubuntu distributions. As of Red Hat Enterprise Linux 8, support was substituted by OpenShift container management platform. In this section, we describe each distribution, while discussing the container landscape on Red Hat.

7.3.1 Containers on Red Hat

With the changes on how containerization is used in the enterprise computing setting, Red Hat dropped support for Docker-related tools and focused on its product OpenShift. The Red Hat OpenShift platform provides Docker functions without exposing Docker tools directly.

Alongside the change, Red Hat developed tools, such as Podman, skopeo, and Buildah, which all assist in configuring and maintaining a container workflow with minimum overhead. These tools provide the following functions:

- ▶ podman: Client tool to manage containers. Replaces most features that are provided by the **docker** command, which focuses on individual containers or images.
- ▶ skopeo: Tool to manage images by copying them to and from registries.
- ▶ runc: Runtime client for running and working with Open Container Initiative (OCI) format.
- ▶ buildah: Tool to manage OCI-compliant images.

For more information about Red Hat OpenShift on IBM Z platform, see [Red Hat OpenShift on IBM Z Installation Guide, REDP-5605](#).

7.3.2 Installing and configuring Docker

The required packages for running Docker containers on Red Hat Linux, SUSE and Ubuntu distributions are available by using different methods.

Installing the Docker service

In this section, we describe installing Docker on each distribution:

- ▶ Red Hat Enterprise Linux
 - Before installing a new version, ensure that any older versions are uninstalled, along with their associated dependencies. Additionally, uninstall Podman and its associated dependencies, if already installed, as shown in Example 7-1.

Example 7-1 Uninstall older version of Docker

```
[root@ylsprd bin]# sudo yum remove docker \
    docker-client \
    docker-client-latest \
    docker-common \
    docker-latest \
    docker-latest-logrotate \
    docker-logrotate \
    docker-engine \
    podman \
    runc
```

```
Updating Subscription Management repositories.
Unable to read consumer identity
```

```
This system is not registered with an entitlement server. You can use
subscription-manager to register.
```

```
No match for argument: docker-client
No match for argument: docker-client-latest
No match for argument: docker-common
No match for argument: docker-latest
No match for argument: docker-latest-logrotate
No match for argument: docker-logrotate
No match for argument: docker-engine
No match for argument: runc
Dependencies resolved.
```

```
=====
Package
```

```
Architecture
```

```

Version                               Repository
Size
=====
Removing:
  podman                               s390x
  2:4.6.1-8.el9_3
  @rhel9-appstream                    53 M
  podman-docker                       noarch
  2:4.6.1-8.el9_3
  @rhel9-appstream                    10 k

Transaction Summary
=====
Remove 2 Packages

Freed space: 53 M
Is this ok [y/N]: y
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :
  1/1
  Erasing   :podman-docker-2:4.6.1-8.el9_3.noarch
  1/2
  RunningScriptlet:podman-2:4.6.1-8.el9_3.s390x
  2/2
  Erasing   :podman-2:4.6.1-8.el9_3.s390x
  2/2
  RunningScriptlet:podman-2:4.6.1-8.el9_3.s390x
  2/2
  Verifying :podman-2:4.6.1-8.el9_3.s390x
  1/2
  Verifying :podman-docker-2:4.6.1-8.el9_3.noarch
  2/2
Installed products updated.

Removed:
  podman-2:4.6.1-8.el9_3.s390x
  podman-docker-2:4.6.1-8.el9_3.noarch

Complete!

[root@ylsprd bin]#

```

- Install the yum-utils package, which provides the **yum-config-manager** utility, and set up the repository as shown in Example 7-2.

Example 7-2 Installing yum-utils package and set repos

```

[root@ylsprd bin]# sudo yum install -y yum-utils
Updating Subscription Management repositories.
Unable to read consumer identity

```

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 1:47:23 ago on Fri 08 Mar 2024 03:20:26 PM CST.

Dependencies resolved.

```
=====
Package                               Architecture
Version                               Repository
Size
=====
Installing:
yum-utils                             noarch
4.3.0-11.el9_3                        rhel9-base
45 k
=====
```

Transaction Summary

Install 1 Package

Total download size: 45 k
Installed size: 23 k
Downloading Packages:
yum-utils-4.3.0-11.el9_3.noarch.rpm
5.9 MB/s | 45 kB 00:00

```
-----
Total
4.3 MB/s | 45 kB 00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
1/1 Installing :yum-utils-4.3.0-11.el9_3.noarch
1/1 Running scriptlet: yum-utils-4.3.0-11.el9_3.noarch
1/1 Verifying :yum-utils-4.3.0-11.el9_3.noarch
1/1 Installed products updated.
```

Installed: yum-utils-4.3.0-11.el9_3.noarch

Complete!

```
[root@y1sprd bin]#
[root@y1sprd bin]# sudo yum-config-manager --add-repo
https://download.docker.com/linux/rhel/docker-ce.repo
Updating Subscription Management repositories.
Unable to read consumer identity
```

This system is not registered with an entitlement server. You can use subscription-manager to register.

Adding repo from: <https://download.docker.com/linux/rhel/docker-ce.repo>

- Proceed to install Docker Engine, containerd, and Docker Compose, as shown in Example 7-3 on page 152.

Example 7-3 Installing Docker engine, contained and compose

```
[root@ylsprd bin]# sudo yum install docker-ce docker-ce-cli containerd.io
docker-buildx-plugin docker-compose-plugin
Updating Subscription Management repositories.
Unable to read consumer identity
```

This system is not registered with an entitlement server. You can use subscription-manager to register.

```
DockerCEStable-s390x
57 kB/s | 3.5 kB    00:00
Dependencies resolved.
```

```
=====
Package
Architecture          Version
Repository              Size
=====
Installing:
containerd.io          s390x
1.6.28-3.1.el9         docker-ce-stable
28 M
docker-buildx-plugin   s390x
0.12.1-1.el9           docker-ce-stable
12 M
docker-ce              s390x
3:25.0.3-1.el9         docker-ce-stable
17 M
docker-ce-cli          s390x
1:25.0.3-1.el9         docker-ce-stable
6.9 M
docker-compose-plugin  s390x
2.24.5-1.el9           docker-ce-stable
12 M
Installing weak dependencies:
docker-ce-rootless-extras s390x
25.0.3-1.el9           docker-ce-stable
4.0 M
```

Transaction Summary

```
=====
Install 6 Packages
```

```
Total download size: 80 M
Installed size: 353 M
Is this ok [y/N]: y
Downloading Packages:
(1/6): docker-buildx-plugin-0.12.1-1.el9.s390x.rpm 27 MB/s | 12 MB
00:00
(2/6): docker-ce-cli-25.0.3-1.el9.s390x.rpm 19 MB/s | 6.9 MB    00:00
(3/6): containerd.io-1.6.28-3.1.el9.s390x.rpm 22 MB/s | 28 MB    00:01
(4/6): docker-ce-rootless-extras-25.0.3-1.el9.s390x.rpm 7.9 MB/s | 4.0 MB
00:00
(5/6): docker-ce-25.0.3-1.el9.s390x.rpm 11 MB/s | 17 MB    00:01
(6/6): docker-compose-plugin-2.24.5-1.el9.s390x.rpm 22 MB/s | 12 MB
00:00
```



```

-----
Total
44 MB/s | 80 MB    00:01
DockerCEStable-s390x
23 kB/s | 1.6 kB   00:00
Importing GPG key 0x621E9F35:
  Userid      : "Docker Release (CE rpm) <docker@docker.com>"
  Fingerprint: 060A 61C5 1B55 8A7F 742B 77AA C52F EB6B 621E 9F35
  From        : https://download.docker.com/linux/rhel/gpg
Is this ok [y/N]: y
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing  :
1/1
Installing :docker-compose-plugin-2.24.5-1.el9.s390x
1/6
Runningscriptlet:docker-compose-plugin-2.24.5-1.el9.s390x
1/6
Installing :docker-buildx-plugin-0.12.1-1.el9.s390x
2/6
Runningscriptlet:docker-buildx-plugin-0.12.1-1.el9.s390x
2/6
Installing :docker-ce-cli-1:25.0.3-1.el9.s390x
3/6
Runningscriptlet:docker-ce-cli-1:25.0.3-1.el9.s390x
3/6
Installing :containerd.io-1.6.28-3.1.el9.s390x
4/6
Runningscriptlet:containerd.io-1.6.28-3.1.el9.s390x
4/6
Installing :docker-ce-rootless-extras-25.0.3-1.el9.s390x
5/6
Runningscriptlet:docker-ce-rootless-extras-25.0.3-1.el9.s390x
5/6
Installing :docker-ce-3:25.0.3-1.el9.s390x
6/6
Runningscriptlet:docker-ce-3:25.0.3-1.el9.s390x
6/6
Verifying :containerd.io-1.6.28-3.1.el9.s390x
1/6
Verifying :docker-buildx-plugin-0.12.1-1.el9.s390x
2/6
Verifying :docker-ce-3:25.0.3-1.el9.s390x
3/6
Verifying :docker-ce-cli-1:25.0.3-1.el9.s390x
4/6
Verifying :docker-ce-rootless-extras-25.0.3-1.el9.s390x
5/6
Verifying :docker-compose-plugin-2.24.5-1.el9.s390x
6/6
Installed products updated.

```

```
Installed:
containerd.io-1.6.28-3.1.el9.s390x  docker-buildx-plugin-0.12.1-1.el9.s390x
docker-ce-3:25.0.3-1.el9.s390x  docker-ce-cli-1:25.0.3-1.el9.s390x
docker-ce-rootless-extras-25.0.3-1.el9.s390x
docker-compose-plugin-2.24.5-1.el9.s390x
```

```
Complete!
[root@y1sprd bin]#
```

- Once installed, start Docker, as shown in Example 7-4.

Example 7-4 Start Docker

```
[root@y1sprd bin]# sudo systemctl start docker
```

► **Ubuntu**

The installation on Ubuntu is straightforward. Because the packages are available on the main repository, they can be immediately installed, as shown in Example 7-5.

Example 7-5 Installing Docker onto Ubuntu

```
root@rdbk86ub:~# apt install docker
```

By running that command, Docker and its related tools are installed.

► **SUSE**

The Docker packages on SUSE are available by way of the Container module. For more information about how to set up that module, see your distribution manual. The installation step is also simple, as shown in Example 7-6.

Example 7-6 Installing Docker on SUSE

```
rdbk86sl:~ # zypper install docker
```

Configuring the Docker service

After the installation is complete, enable and start the Docker engine. The same command is applied to Ubuntu and SUSE, but we used SUSE in our Example 7-7.

Example 7-7 Enabling and starting Docker service

```
rdbk86sl:~ # systemctl enable --now docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service ?
/usr/lib/systemd/system/docker.service.
rdbk86sl:~ #
```

Setting up user access to Docker

Because the Docker daemon binds to a UNIX socket instead of a TCP port, the Docker daemon always runs as a root user. To avoid having to use **sudo** when Docker commands are used, add the user to the Docker group, as shown in Example 7-8.

Example 7-8 Add unprivileged user to Docker group

```
rdbk86sl:~ # usermod -aG docker lnxadmin
rdbk86sl:~ #
```

Log off and log in again for the user privilege to take effect. You can then verify the Docker commands, as shown in Example 7-9.

Example 7-9 Verifying Docker version

```
lnxadmin@rdbk86sl:~> docker --version
Docker version 19.03.11, build 42e35e61f352
lnxadmin@rdbk86sl:~>
```

7.3.3 Testing Docker

As part of the verification, access the pre-built Docker images that are part of the Docker Hub and run a test hello-world container, as shown in Example 7-10.

Example 7-10 Running hello-world container

```
[root@ylsprd bin]# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
f19d1e240d64: Pull complete
Digest: sha256:d000bc569937abbe195e20322a0bde6b2922d805332fd6d8a68b19f524b7d21d
Status: Downloaded newer image for hello-world:latest
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(s390x)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

<https://hub.docker.com/>

For more examples and ideas, visit:

<https://docs.docker.com/get-started/>

```
[root@ylsprd bin]# docker image list
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    37fc523148ea   10 months ago  9.06kB
```

You successfully deployed Docker.

7.4 Deploying MongoDB on Linux on IBM Z

MongoDB is an open source database that is considered to be a popular and fast growing NoSQL database. This is mostly due to how well it works in areas where traditional SQL databases have trouble. It is good for dealing with large sets of unstructured data and has exceptionally good read times on the data that is stored. Although it is not a replacement for all SQL applications that store structured data, it does give a modern solution for the massive amounts of unstructured data and mobile traffic.

With the performance and virtualization capabilities of Linux on IBM Z, it makes an ideal platform for scaling out and scaling up MongoDB based NoSQL workloads.

To install MongoDB on Linux on IBM Z, you have several options:

- ▶ Many Linux distributions provide MongoDB packages in their official repositories. You can use your distribution's package manager (such as apt for Ubuntu-based systems or yum for Red Hat/CentOS-based systems) to install MongoDB.
- ▶ MongoDB provides official repositories for various Linux distributions. You can add the MongoDB repository to your system and then install MongoDB using your package manager. This method allows you to install the latest stable version of MongoDB.
- ▶ Alternatively, you can download the MongoDB binaries from the MongoDB website and manually install them on your system. This method gives you more control over the installation process but requires you to handle dependencies and configuration manually.
- ▶ Another option is to use containerization tools like Docker or Podman to run MongoDB in a containerized environment. This approach isolates MongoDB and its dependencies from the host system and simplifies deployment and management.

In this section, we cover both the steps of installing MongoDB by using Docker as well as the steps to install Package Manager.

7.4.1 Deploying MongoDB as a Docker container

In this section, we outline and discuss deploying MongoDB as a Docker container.

Our lab environment

This example uses Ubuntu 20.04 LTS as the host operating system for the MongoDB deployment. Because we decided to install MongoDB as a Docker container, the first step is to set up Docker on the host systems. For more information about deploying Docker on Ubuntu, see *The Virtualization Cookbook for IBM z Systems Volume 4: Ubuntu Server 16.04*, SG24-8354.

Important: The Docker installation package available in the official Ubuntu 20.04 repository might not be the latest version. To get the latest version, install Docker from the official Docker repository.

After Docker is configured, enable its service and run it on the host operating system. Example 7-11 shows verifying the Docker configuration.

Example 7-11 Verification of Docker

```
tnxadmin@rdbk86ub:~$ docker version
Client:
Version:           19.03.8
```

```

API version:      1.40
Go version:       go1.13.8
Git commit:       afacb8b7f0
Built:           Tue Jun 23 22:26:11 2020
OS/Arch:         linux/s390x
Experimental:    false

```

Server:

```

Engine:
  Version:        19.03.8
  API version:    1.40 (minimum version 1.12)
  Go version:     go1.13.8
  Git commit:     afacb8b7f0
  Built:         Thu Jun 18 08:26:54 2020
  OS/Arch:       linux/s390x
  Experimental:   false

```

```

containerd:
  Version:        1.3.3-0ubuntu2
  GitCommit:
runc:
  Version:        spec: 1.0.1-dev
  GitCommit:
docker-init:
  Version:        0.18.0
  GitCommit:

```

```
1nxadmin@rdbk86ub:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED
SIZE			

```
1nxadmin@rdbk86ub:~$
```

IBM has been working on containerizing important open source products and tools for its various platforms and also making them available on Docker Hub public registry for download. Docker Hub is a cloud-based registry service that allows you to link to code repositories, build your images and test them, and store manually pushed images and links to Docker Cloud so you can deploy images to your hosts. It provides a centralized resource for container image discovery, distribution, and change management.

Run the **docker search** command to search for repositories specific to a platform in Docker Hub, as shown in Example 7-12. The command returns the pre-built Docker images for Linux on IBM Z from the public registry.

Example 7-12 Pre-built Docker Images for Linux on IBM Z

```

1nxadmin@rdbk86ub:~$ docker search --filter=is-official=true mongodb
NAME                DESCRIPTION                                STARS
OFFICIAL            AUTOMATED
mongo               MongoDB document databases provide high avai...  7227
[OK]
mongo-express       Web-based MongoDB admin interface, written w...  788
[OK]
1nxadmin@rdbk86ub:~$

```

MongoDB container deployment

Now that you have a pre-built image for MongoDB from the Docker Hub, run commands to Docker to download and register the image to the local host system, as shown in Example 7-13. These images are read-only snapshots of defined layers and commands.

Example 7-13 Downloading Linux on IBM Z MongoDB Image from Docker Hub

```
lnxadmin@rdbk86ub:~$ docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
dd2de95b9a1c: Pull complete
c38a48ef4dfa: Pull complete
eced51184728: Pull complete
a9288641caad: Pull complete
bf72d8578ae0: Pull complete
891f80311986: Pull complete
b862844c0ef4: Pull complete
185ef58863de: Pull complete
36be598b0fb9: Pull complete
afd583fd9ef0: Pull complete
Digest: sha256:a4448eb5f6e6097353d0ab97eb50aeb0238bb4e60c37e401920d3c2c4fc73eb9
Status: Downloaded newer image for mongo:latest
docker.io/library/mongo:latest
lnxadmin@rdbk86ub:~$
```

Verify that the image was correctly registered with the local Docker registry and allocated a local image ID, as shown in Example 7-14.

Example 7-14 Verification of MongoDB Docker image pull

```
lnxadmin@rdbk86ub:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED
mongo	latest	abc0c0551238	2 weeks ago

```
491MB
lnxadmin@rdbk86ub:~$
```

When the MongoDB container was built, the directories `/data/configdb` and `/data/db` were used as mount points for external storage, and it exposes ports 27017 and 28017. This technique allows connections from outside the container to access the MongoDB container. Example 7-15 shows the configuration.

Example 7-15 Docker inspect

```
lnxadmin@rdbk86ub:~$ docker inspect mongo
[
  {
    "Id":
"sha256:abc0c05512382652300ed10d809912071b8ee0ca14c10891cabccd8228c6dc94",
    "Created": "2020-09-25T23:30:04.442106691Z",
    ...
    "Image": "mongo/mongo",
    "Volumes": {
      "/data/configdb": {},
      "/data/db": {}
    },
  },
]
```

```
},
```

As shown in Example 7-15 on page 158, the pre-built MongoDB container stores the data on the `/data/` folder on the host system. The idea is to create a data directory on the host system (outside of the container) and mount this to a directory visible from inside the container. This configuration places the database files in a known location on the host system, and makes it easy for tools and applications on the host system to access the files. Create a folder on the host system as shown in Example 7-16.

Example 7-16 Create data directory

```
lnxadmin@rdbk86ub:~$ sudo mkdir -p /data/db
lnxadmin@rdbk86ub:~$ sudo mkdir -p /data/configdb
```

Running MongoDB container

Starting the MongoDB container by running the **docker run** command. Example 7-17 shows that the Docker should instantiate the image named `mongo/mongo` and assign the newly instantiated container with the name `itsomongo`, while mounting local volumes to serve as persistent storage. Using this technique allows you to refer to the container by name rather than having to use the ID hash. If you do not provide a name, Docker assigns one from some randomly selected words. Also, specify the ports so that it maps the default MongoDB port 27017 to an external port.

The response to the successful instantiation would be a return of a hash that is the full ID of the new running container.

Example 7-17 Starting MongoDB container

```
lnxadmin@rdbk86ub:~$ docker run -p 27017:28017 --name itsomongo --mount
type=bind,source=/data/configdb/,target=/data/configdb --mount
type=bind,source=/data/db/,target=/data/db -d mongo
c440495d1e03ba855561b03e6114ee6dc2efe68d412b4ac9bc181a74ce29e75d
lnxadmin@rdbk86ub:~$
```

Verifying and accessing MongoDB container

Check whether the container named `itsomongo` has started by running the **docker ps** command, as shown in Example 7-18. The status column of the command output shows that the MongoDB container is up and already listening. In addition, the output provides the mapping of 27017 and 28017 as the container's local ports.

Example 7-18 Container startup verification

```
lnxadmin@rdbk86ub:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND NAMES	CREATED	STATUS
c440495d1e03	mongo	"docker-entrypoint.s..."	8 seconds ago	Up 7
seconds	27017/tcp, 0.0.0.0:27017->28017/tcp	itsomongo		

```
lnxadmin@rdbk86ub:~$
```

For more information about the container, inspect the container by running the **docker inspect <container id>** command.

You can use the following methods to access MongoDB from the host:

- ▶ Use MongoDB client tools
- ▶ Use Docker to connect to the MongoDB container shell and verify the database

This example uses the latter option. Therefore, start a Docker interactive shell into the MongoDB container and start a Mongo shell for creating a sample database (see Example 7-19).

Example 7-19 Access MongoDB Container using Docker

```
lnxadmin@rdbk86ub:~$ docker exec -it itsomongo /bin/bash
root@c440495d1e03:/# mongo
MongoDB shell version v4.4.1
connecting to:
mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("3dd136fc-e749-469a-a6b4-61c178184e9d") }
MongoDB server version: 4.4.1
Welcome to the MongoDB shell.
For more comprehensive documentation, see
    https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
    https://community.mongodb.com
> use mongodb
switched to db mongodb
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
> db.itso.insert({"Redbook":"Linux on IBM Z"})
WriteResult({ "nInserted" : 1 })
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
mongodb  0.000GB
>
```

This method provides a quick way to have a highly scalable environment to work on for any solution that involves MongoDB containers that are deployed on Linux on IBM Z.

Migrating MongoDB data

As with any other database, MongoDB offers different options to migrate data. The most straightforward way of moving data between installations is to dump and import. This methods means generating a dump on the source system, transferring it to the target system, then importing it from scratch.

Attention: This method might not be suited for all installations and environments, depending on database size or other constraints. Consult the MongoDB manuals for more options about migrating data.

Generating a dump from the source system

To generate a dump with the source system's data, you can run the **mongodump** command. In our example, we connect to the source system `xrhrbres2` and run the command to generate the dump file, as shown in Example 7-20.

Example 7-20 Dumping data from the source system

```
root@xrhrbres2:/data# mongodump --out /data/db/backup/`date +%y-%m-%d`~
```



```
2020-10-23T15:06:44.077+0000    writing mongodb.itso to
/data/db/backup/20-10-23/mongodb/itso.bson
2020-10-23T15:06:44.078+0000    done dumping mongodb.itso (1 document)
root@xrhrbres2:/data#
```

As shown in Example 7-20 on page 160, the files were exported to path `/data/db/backup/20-10-23/`. We now can tar and compress the file, moving it to our target server `rdbk86ub` after using `rsync`. (see Example 7-21).

Example 7-21 Compressing and transferring the dump to the target system

```
root@xrhrbres2:/data/db/backup# tar -jcf 20-10-23.tar.bz2 20-10-23
root@xrhrbres2:/data/db/backup# rsync -v 20-10-23.tar.bz2
lnxadmin@rdbk86ub.pbm.ihost.com:/data/
root@xrhrbres2:/data/db/backup#
```

Importing data into the target system

On the target system, because we use Docker to run MongoDB, we must place the file on a volume that is exported to the Docker container. In our case, we also are using `/data/db/backup` on the target system. After the file is transferred, we extract the data to prepare for importing it onto MongoDB, as shown in Example 7-22.

Example 7-22 Extracting data to be imported

```
lnxadmin@rdbk86ub:/data/db/backup$ tar -xf 20-10-23.tar.bz2
lnxadmin@rdbk86ub:/data/db/backup$ ls -l
total 8
drwxr-xr-x 3 lnxadmin lnxadmin 4096 Oct 23 11:06 20-10-23
-rw-r--r-- 1 lnxadmin lnxadmin 385 Oct 23 11:11 20-10-23.tar.bz2
lnxadmin@rdbk86ub:/data/db/backup$
```

We can now import data onto MongoDB. In this case, we must open a shell to the active container and then, run the `mongorestore` command to import the data, as in Example 7-23.

Example 7-23 Importing data into the deployed container

```
lnxadmin@rdbk86ub:/data/db/backup$ docker exec -it itsomongo /bin/bash
root@c440495d1e03:/data/db/backup# mongorestore --db mongodb --drop
/data/db/backup/20-10-23/mongodb/
2020-10-23T15:25:59.978+0000    building a list of collections to restore from
/data/db/backup/20-10-23/mongodb dir
2020-10-23T15:25:59.980+0000    reading metadata for mongodb.itso from
/data/db/backup/20-10-23/mongodb/itso.metadata.json
2020-10-23T15:25:59.986+0000    restoring mongodb.itso from
/data/db/backup/20-10-23/mongodb/itso.bson
2020-10-23T15:25:59.987+0000    no indexes to restore
2020-10-23T15:25:59.987+0000    finished restoring mongodb.itso (1 document, 0
failures)
2020-10-23T15:25:59.987+0000    1 document(s) restored successfully. 0 document(s)
failed to restore.
root@c440495d1e03:/data/db/backup#
```

After the import is done, we can select the data from the imported dump file, as shown in Example 7-24 on page 162.

Example 7-24 Selecting data after import

```
lnxadmin@rdbk86ub:~$ docker exec -it itsomongo /bin/bash
root@c440495d1e03:/# mongo
> use mongodb
switched to db mongodb
> db.itso.find()
{ "_id" : ObjectId("5f92f5383d2effdcc886b655"), "Redbook" : "Linux on IBM Z" }
>
```

In this example, the database that was created on a x86 server installation was moved to a Linux on IBM Z server that is running MongoDB as a container without any issues or special processes.

7.4.2 Deploying MongoDB by using package manager

In this section, we outline and discuss deploying MongoDB by using package manager.

Work environment

This example uses Red Hat Enterprise Linux 9.3 as the host operating system for the MongoDB deployment. We will use package manager to install MongoDB on the host system.

MongoDB installation and configuration

- Install essential development tools and libraries using **yum**. Install the following packages: **gcc**, **cpp**, **make**, **autoconf**, **automake**, **libtool**, **openssl**, **openssl-devel**, **bzip2-devel**, **libcurl-devel**, **libstdc++-devel**, **systemd**, and **systemd-libs** as shown in Example 7-25.

Example 7-25 Installing dependencies

```
[root@redbooks1 linux1]# yum install cyrus-sasl cyrus-sasl-plain
cyrus-sasl-gssapi krb5-libs lm_sensors-libs net-snmp-agent-libs net-snmp
openssl rpm-libs
```

- Install the **compat-openssl11** package as shown in Example 7-26.

Example 7-26 Installing compat-openssl11 package

```
[root@redbooks1 linux1]# rpm -ivh compat-openssl11-1.1.1k-4.el9.s390x.rpm
warning: compat-openssl11-1.1.1k-4.el9.s390x.rpm: Header V3 RSA/SHA256 Signature, key ID
8483c65d: NOKEY
Verifying... ##### [100%]
Preparing... ##### [100%]
Updating / installing...
 1:compat-openssl11-1:1.1.1k-4.el9 ##### [100%]
[root@redbooks1 linux1]#
```

- Go to the MongoDB [download page](#) and select the appropriate MongoDB Enterprise package for your Red Hat/CentOS 7.0.6 and architecture (S390).
- Once downloaded, use the **rpm** package manager to install the MongoDB Enterprise RPM package as shown in Example 7-27.

Example 7-27 Installing MongoDB RPM

```
[root@redbooks1 linux1]# rpm -ivh mongodb-enterprise-server-7.0.6-1.el8.s390x.rpm
```

```
warning: mongodb-enterprise-server-7.0.6-1.el8.s390x.rpm: Header V4 RSA/SHA256 Signature, key ID
1785ba38: NOKEY
Verifying... ##### [100%]
Preparing... ##### [100%]
Updating / installing...
 1:mongodb-enterprise-server-7.0.6-1##### [100%]
Created symlink /etc/systemd/system/multi-user.target.wants/mongod.service ?
/usr/lib/systemd/system/mongod.service.
```

- Verify MongoDB version as shown in Example 7-28.

Example 7-28 Verify MongoDB version

```
[root@redbooks1 lib]# mongod --version
db version v7.0.6
Build Info: {
  "version": "7.0.6",
  "gitVersion": "66cdc1f28172cb33ff68263050d73d4ade73b9a4",
  "opensslVersion": "OpenSSL 1.1.1k 25 Mar 2021",
  "modules": [
    "enterprise"
  ],
  "allocator": "tcmalloc",
  "environment": {
    "distmod": "rhel83",
    "distarch": "s390x",
    "target_arch": "s390x"
  }
}
[root@redbooks1 lib]#
```

- Enable access to 27017 ports for SELinux if using enforcing mode as shown in Example 7-29.

Example 7-29 Enable access to MongoDB port

```
[root@redbooks1 linux1]# semanage port -a -t mongod_port_t -p tcp 27017
```

- Start the mongod process as shown in Example 7-30.

Example 7-30 Start MongoDB

```
[root@redbooks1 linux1]# sudo service mongod start
```

- The MongoDB Shell (mongosh) is a handy tool for interacting with your MongoDB database. Download the appropriate MongoDB Shell RPM package from the MongoDB website and install it using rpm as shown in Example 7-31.

Example 7-31 Download and install MongoDB Shell

```
[root@redbooks1 mongosh]# wget
https://downloads.mongodb.com/compass/mongosh-2.1.5-linux-s390x.tgz
--2024-03-04 14:15:26--
https://downloads.mongodb.com/compass/mongosh-2.1.5-linux-s390x.tgz
Resolving downloads.mongodb.com (downloads.mongodb.com)... 18.164.96.33,
18.164.96.65, 18.164.96.74, ...
Connecting to downloads.mongodb.com (downloads.mongodb.com)|18.164.96.33|:443...
connected.
```

```
HTTP request sent, awaiting response... 200 OK
Length: 60269129 (57M) [application/octet-stream]
Saving to: 'mongosh-2.1.5-linux-s390x.tgz'
```

```
mongosh-2.1.5-linux-s390x.tgz
100%[=====] 57.48M
17.9MB/s in 3.2s
```

```
2024-03-04 14:15:29 (17.9 MB/s) - 'mongosh-2.1.5-linux-s390x.tgz' saved
[60269129/60269129]
```

```
[root@redbooks1 mongosh]# ls
mongosh-2.1.5-linux-s390x.tgz
[root@redbooks1 mongosh]# gunzip mongosh-2.1.5-linux-s390x.tgz
[root@redbooks1 mongosh]# tar xvf mongosh-2.1.5-linux-s390x.tar
```

-
- ▶ Connect to MongoDB Shell (mongosh) and create a new database as shown in Example 7-24 on page 162.

Example 7-32 Create new database

```
./mongosh
Current Mongosh Log ID: 65e62c2f8f4d71346332d86c
Connecting to:
mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&app
Name=mongosh+2.1.5
Using MongoDB: 7.0.6
Using Mongosh: 2.1.5
```

For mongosh info see: <https://docs.mongodb.com/mongosh-shell/>

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (<https://www.mongodb.com/legal/privacy-policy>). You can opt-out by running the `disableTelemetry()` command.

```
-----
The server generated these startup warnings when booting
2024-03-04T11:43:50.655-06:00: Using the XFS filesystem is strongly recommended
with the WiredTiger storage engine. See
http://dochub.mongodb.org/core/prodnotes-filesystem
2024-03-04T11:43:50.983-06:00: Access control is not enabled for the database.
Read and write access to data and configuration is unrestricted
2024-03-04T11:43:50.983-06:00: /sys/kernel/mm/transparent_hugepage/enabled is
'always'. We suggest setting it to 'never'
2024-03-04T11:43:50.983-06:00: vm.max_map_count is too low
-----
```

```
Enterprise test>
Enterprise test> use ysldb
switched to db ysldb
Enterprise ysldb> show dbs
admin 40.00 KiB
config 60.00 KiB
local 72.00 KiB
```

Migrating MongoDB data

Use the same procedure as explained in “Migrating MongoDB data” on page 160.

7.5 Deploying MediaWiki and MySQL

With the Linux operating system comes a wide variety of open-source applications. A very popular application for Linux is MediaWiki, the general-purpose wiki that originated with Wikipedia. It is written in PHP and uses MySQL as its backend database. This configuration is commonly known as a LAMP server, meaning that the application employs Linux, Apache, MySQL, and PHP. This Web 2.0 stack is an ideal workload for Linux on IBM Z.

In this example, the MySQL database is contained on its own disk partition on an external iSCSI disk. Likewise, the DocumentRoot of the Apache webserver is also stored on its own external iSCSI disk, different than the LUN where the MySQL database is stored.

Storing application data on iSCSI external disks is a popular method because makes it much easier to move (or migrate) services from one host to another. It commoditizes the operating system and the various services. The infrastructure can easily be adapted and scaled to meet demand, while keeping the data available universally from a main storage system.

In this example we have successfully used MySQL as a back end to MediaWiki. The MediaWiki documentation describes a method to use MariaDB as the backend database.

7.5.1 Analysis and planning

Following the guidelines and recommendations outlined in Chapter 4, “Migration process” on page 49, and Chapter 6, “Migration analysis” on page 71, appropriate planning and analysis should be performed before these migration activities. The checklists are helpful in identifying how virtual resources should be dedicated and organized.

For this example in our lab environment, the z/VM guest has already been set up and a minimal Linux operating system installed. A Red Hat Enterprise Linux Enterprise (RHEL) 9.3 guest, named rdbkpmr1 with one virtual CPU, and 16 GB of virtual memory is installed. It is presumed that an adequate package management (RPM) repository for installation source is already set up and available for the installation of the application software that will be used. We will be installing the LAMP stack, as well as MediaWiki.

The Linux environment on x86 is largely the same as it is on IBM Z (with a few notable exceptions). Configuration files on the x86 will be in the same place on your Linux guest on IBM Z, unless you deliberately choose to keep them in a different place. Hence, the MySQL configuration files, for example, will typically only need to be copied from the x86 server to the Linux on IBM Z server, placed in the same location in the file system, `/etc/my.cnf`.

Best practices dictate that migrating to IBM Z be performed first to a test environment. Then, after successfully testing the deployment in the test environment, migration to the production environment is appropriate.

7.5.2 Installing the LAMP stack on Red Hat Enterprise Linux

Log into the rdbkpmr1 server.

DaNdiFied yum (DNF) is the Yum package manager for RHEL, and can be updated with the following command:

```
sudo dnf update
```

Install Apache HTTP Daemon (httpd) by using the following command:

```
sudo dnf install httpd
```

Install MySQL by using the following command:

```
sudo dnf install mysql-server
```

Install PHP by using the following command:

```
sudo dnf install @php
```

7.5.3 Starting and testing LAMP components

The Apache and MySQL configurations in this example scenario are simple, whereas your configuration may be more complex. In our example, MediaWiki is the only application configured for Apache and no other data exists in the MySQL database than what is used by MediaWiki.

Although version information about the packages may have been noticed while installing the packages, it is helpful to confirm that the version of Apache is what is expected. A common method of displaying the version is by running **apachectl -v**.

Example 7-33 shows the command and resulting output that you would use that to display the version of Apache.

Example 7-33 Output of apachectl -v

```
[lnxadmin@rdbk86r1 bin]# apachectl -v
Server version: Apache/2.4.57 (Red Hat Enterprise Linux)
Server built: Jul 20 2023
```

Historically, it was common to have the installed services started automatically when the package was installed. Today, it is more common that the distribution takes a more active role in ensuring that potentially renegade software does not start automatically. Hence, it is necessary to start Apache manually, and to set it to start automatically each time the system is booted.

Apache services on Red Hat Enterprise Linux

Start the apache web service by using the following commands:

```
sudo systemctl start httpd
sudo systemctl enable httpd
```

Verify that the web server is running

With the web service started, a web browser should be used to verify that the web server is actually working as expected, and as shown in Figure 7-1 on page 167. Start a web browser and point it to the IP address of your Linux server, in our case, the URL was `http://129.40.23.229`.

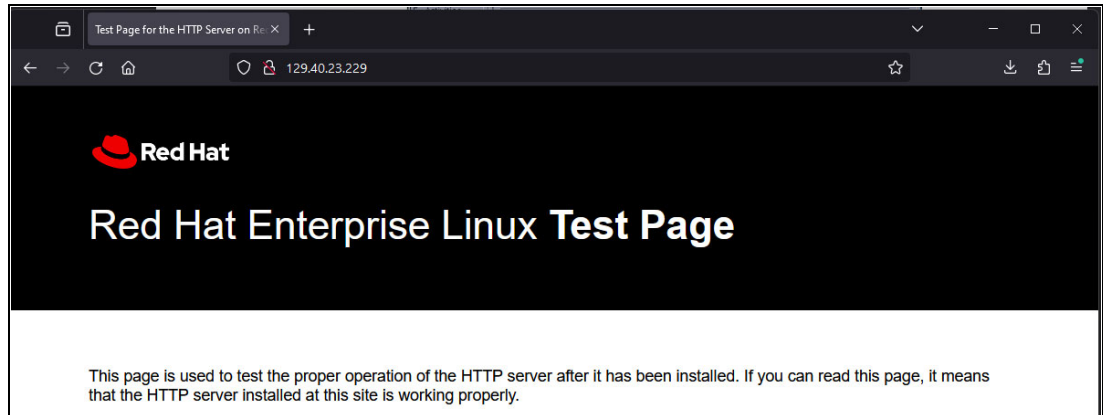


Figure 7-1 Successful test of Apache installation

You may need to open the firewall in order to access the web server. The following commands will accomplish this task:

```
sudo firewall-cmd --permanent --zone=public --add-service=http
sudo firewall-cmd --permanent --zone=public --add-service=https
sudo firewall-cmd --reload
```

Verify that PHP is working

Before a test can be conceived and executed for PHP, the location of the DocumentRoot directory of the Apache server must be determined.

Under RHEL, the default location is `/var/www/html`, but the definitive value can be found by using the command shown in Example 7-34.

Example 7-34 Finding the DocumentRoot on RHEL

```
[lnxadmin@rdbk86r1 ~] grep 'DocumentRoot "' /etc/httpd/conf/httpd.conf
DocumentRoot "/var/www/html"
```

After confirming the Document Root of the Apache server, a one-line PHP script is created that will print the standard PHP installation information. Using `vi` or some other appropriate text editor, create a script file called `phpinfo.php`, as shown in Example 7-35, and place the script file in the appropriate DocumentRoot directory.

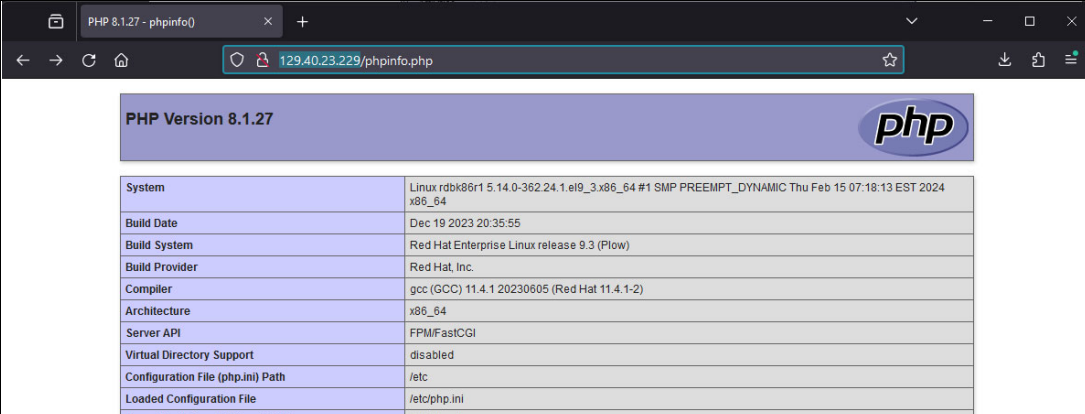
Example 7-35 Simple PHP script that displays functional characteristics

```
<?php phpinfo(); ?>
```

With the PHP script file in the DocumentRoot directory, the PHP script can be run by using a web browser. Connect to your web server, using your URL and the script name, as an example:

```
http://129.40.23.229/phpinfo.php
```

Figure 7-2 on page 168 shows the expected PHP information that is generated in the browser by the PHP script running on RHEL.



PHP Version 8.1.27	
System	Linux rdbk86r1 5.14.0-362.24.1.el9_3.x86_64 #1 SMP PREEMPT_DYNAMIC Thu Feb 15 07:18:13 EST 2024 x86_64
Build Date	Dec 19 2023 20:35:55
Build System	Red Hat Enterprise Linux release 9.3 (Plow)
Build Provider	Red Hat, Inc.
Compiler	gcc (GCC) 11.4.1 20230605 (Red Hat 11.4.1-2)
Architecture	x86_64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini

Figure 7-2 PHP configuration information generated by `phpinfo.php`

Start MySQL services on Red Hat Enterprise Linux

To set MySQL to start each time the server is booted, and to manually start the service, issue the following commands:

```
sudo systemctl start mysqld.service
sudo systemctl enable mysqld.service
```

Verify that MySQL is working

MySQL must be configured and working properly before MediaWiki can even be installed. The following are the two configuration steps to complete for MySQL.

1. Copy a sample configuration file to MySQL's production configuration, `/etc/my.cnf`. Then, apply the appropriate ownership and access. (Reading through the configuration file to understand its contents is a wise thing to do.) Example 7-36 shows sample commands.

Later, when migrating from the x86 server, you will likely copy the `my.cnf` file from the x86 server to Linux on IBM Z. For now, the example `my.cnf` configuration file is sufficient in order to test the functionality of the system before migrating.

Example 7-36 Configure MySQL configuration file

```
cp /usr/share/mysql/my-medium.cnf /etc/my.cnf
chown root:root /etc/my.cnf
chmod 640 /etc/my.cnf
```

Note: The default permissions of `/etc/my.cnf` are 644, allowing anyone to read the MySQL configuration settings. Best practices in security suggest that system services should not provide any unnecessary information to unprivileged users. Setting the permissions to 640 prevents unprivileged users from discovering information about the configuration of the MySQL server.

2. Set a temporary password for the database administrator. Remember this password because it is required during a few additional steps of the process before migrating the MediaWiki application from the x86 server. (This may or may not be the same password of the MySQL database that will later be migrated.) Use the command shown in Example 7-37 to set the password.

Example 7-37 Set administrative password for the MySQL service

```
mysqladmin -u root password agoodpassword
```


With the admin password set for the root user, all future interactions with the MySQL database will require providing a password. General administrative functions will require the root password, whereas commands involving MediaWiki will use a different password.

Note: Quotation marks in Linux can be a bit tricky. When setting the root password, keep in mind that the quotation marks are not strictly necessary. If the password will contain special characters, such as a space, then the quotation marks are necessary. Do not use quotations marks unless you are certain that they are necessary. Copying a string from somewhere and pasting the string as the password can give unexpected results, and may make reproducing the password later an inconvenient mystery.

Test the MySQL capabilities by running the following sample command:

```
mysql -u root -p -e "show tables" mysql
```

The preceding command will prompt you for the root password that you set in the previous steps with the `mysqladmin` command. The sample output displayed in Example 7-38 shows the list of tables contained in the MySQL database, suggesting that you have properly set the password.

Example 7-38 Output from the “show tables” mysql command after providing password

```
[root@rdbk86r1 mysql]# mysql -u root -p -e "show tables" mysql
Enter password:
```

```
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| component       |
| db              |
| default_roles   |
| engine_cost     |
| func            |
| general_log     |
| global_grants   |
| gtid_executed   |
| help_category   |
| help_keyword    |
| help_relation   |
| help_topic      |
| innodb_index_stats |
| innodb_table_stats |
| password_history |
| plugin          |
| procs_priv      |
| proxies_priv    |
| replication_asynchronous_connection_failover |
| replication_asynchronous_connection_failover_managed |
| replication_group_configuration_version |
| replication_group_member_actions |
| role_edges      |
| server_cost     |
| servers         |
| slave_master_info |
| slave_relay_log_info |
| slave_worker_info |
+-----+
```

```
| slow_log  
| tables_priv  
| time_zone  
| time_zone_leap_second  
| time_zone_name  
| time_zone_transition  
| time_zone_transition_type  
| user
```

+-----+

With the MySQL administrative password properly set, it is now possible to proceed to installing the MediaWiki software. If the MySQL administrative password has been set up incorrectly, an error message similar to Example 7-39 is displayed.

Example 7-39 Bad password supplied to MySQL

```
[lnxadmin@rdbk86r1 ~]$ mysql -u root -p -e "show tables" mysql  
Enter password:  
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password:  
YES)
```

To correct this issue, run the `mysqladmin` command again as shown in Example 7-37 on page 168, taking extra care to set the password to a value that you will remember. If the original password cannot be remembered or is otherwise lost, it will be necessary to reinstall MySQL.

Note: Another migration option for MySQL is to run `mysqldump` on the x86 server, transfer the resulting MySQL dump files to the Linux on IBM Z, and restore the database with `mysqlimport`. This is generally a recommended and widely used practice, but your environment may dictate a different practice. Your proper pre-migration analysis will help you to understand which approach is best for your circumstances.

With the preliminary Apache, MySQL, and PHP configurations functioning properly on the new Linux on IBM Z, the iSCSI disks can now be migrated from the x86 server.

7.5.4 Installing MediaWiki

The installation instructions for MediaWiki can be found at [Manual:Running MediaWiki on Red Hat Linux](#).

7.5.5 Migrating iSCSI disks containing MySQL and MediaWiki

One of the particularly useful aspects of using external storage is that disks can effectively be unplugged from one server and plugged into another, fast-tracking the migration process. This is not an appropriate approach for all cases, but it is very exciting when circumstances exist that allow this approach. Such is the case for this MediaWiki migration example.

Recall from the original explanation of the scenario that the MySQL database and the Apache DocumentRoot are each self-contained on their own iSCSI disk partitions. The file systems on those partitions will be unmounted from the rdbk86r1 host (an x86 system), then mounted on the rdbkpmr1 host running on IBM Z.

Note: When dealing with remote disk storage, it is important to mount and manage the remote LUNs by referring to them by-path rather than any other method. udev will not ensure that the same name or ID will be used persistently when the host is rebooted. The only persistent identification is by-path.

Prepare Linux on IBM Z for iSCSI

Before making any changes on the x86 host that is currently using the iSCSI LUNs, the Linux guest running on IBM Z should have the minimum iSCSI software already setup:

1. Connect to the Linux on IBM Z guest called rdbkpmr1 using Secure Shell (SSH).
2. Ensure that the iSCSI initiator software is installed on Linux for IBM Z guest rdbkpmr1:

```
sudo dnf install iscsi-initiator-utils
```

3. Stop Apache and MySQL services running on rdbkpmr1. This guest has been running Apache and MySQL for the earlier tests. Stopping these services now is important for the migration.

```
service httpd stop
service mysqld stop
```

4. Move the content of the /srv/www directory and the /var/lib/mysql directory out of the way so that the file systems of the iSCSI remote LUNs can be mounted in their places. But keep their content available as a backup. (Remember that the default DocumentRoot for RHEL is /var/lib/html/. Use the proper directory for your circumstances.)

```
mv /var/www /srv/www.orig
mv /var/lib/mysql /var/lib/mysql.orig
mkdir -p /srv/www /var/lib/mysql
```

Prepare x86 system for migration

1. Connect to the rdbk86r1 (x86) host using SSH.
2. Display the mount table of rdbk86r1, taking note of which file system contains the MySQL partition and the Apache www partition:

mount

In this example, the /var/lib/mysql directory is mounted via /dev/sdd1 and the /srv/www directory is mounted via /dev/sdc1. Example shows a similar method of displaying the mounted file systems by using **df -h**.

Example 7-40 Mounted disk partitions on rdbk86r1

[lnxadmin@rdbk86r1]\$ df -h					
Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	4.0M	0	4.0M	0%	/dev
tmpfs	16G	24K	16G	1%	/dev/shm
tmpfs	6.3G	9.5M	6.3G	1%	/run
/dev/mapper/rhel-root	180G	19G	162G	11%	/
/dev/vda1	2.0G	337M	1.7G	17%	/boot
tmpfs	3.2G	116K	3.2G	1%	/run/user/1000
tmpfs	3.2G	52K	3.2G	1%	/run/user/42
/dev/sda	126G	24K	120G	1%	/mnt/iscsiTest

3. Take note of which remote LUNs are currently being mounted on rdbk86r1-u1:

```
ls -l /dev/disk/by-path/
```

Look for symlinks in the directory that has the following format:

<IPADDRESS>:3260-iscsi-iqn.<iSCSI_Target_Identifier>:<iSCSI_unique_LUN>

Note which remote LUN will be moved, and what symlink it is linked to. In this example, the remote by-path partition for the Apache www disk is identified as the following file system object:

ip-129.40.23.230:3260-iscsi-iqn.2024-03.rdbk86r1.pbm.ihost.com:lun1-lun-1

where:

- The IP address of the remote iSCSI target = 129.40.23.230
- The iSCSI Target Identifier = iscsi-iqn.2024-03.rdbk86r1.pbm.ihost.com
- The unique iSCSI LUN for the Apache www partition = LUN1

Example 7-41 shows the by-path allocations for this example.

Example 7-41 Output showing the by-path assignments of remote iSCSI disks

```
[lnxadmin@rdbk86r1 www]$ ls -l /dev/disk/by-path/
total 0
lrwxrwxrwx. 1 root root 9 Mar 22 11:20
ip-129.40.23.230:3260-iscsi-iqn.2024-03.rdbk86r1.pbm.ihost.com:lun1-lun-1 ->
../../sda
lrwxrwxrwx. 1 root root 9 Mar 18 13:06 pci-0000:00:1f.2-ata-1 -> ../../sr0
lrwxrwxrwx. 1 root root 9 Mar 18 13:06 pci-0000:00:1f.2-ata-1.0 -> ../../sr0
lrwxrwxrwx. 1 root root 9 Mar 18 13:06 pci-0000:04:00.0 -> ../../vda
lrwxrwxrwx. 1 root root 10 Mar 18 13:06 pci-0000:04:00.0-part1 -> ../../vda1
lrwxrwxrwx. 1 root root 10 Mar 18 13:06 pci-0000:04:00.0-part2 -> ../../vda2
lrwxrwxrwx. 1 root root 9 Mar 18 13:06 virtio-pci-0000:04:00.0 -> ../../vda
lrwxrwxrwx. 1 root root 10 Mar 18 13:06 virtio-pci-0000:04:00.0-part1 ->
../../vda1
lrwxrwxrwx. 1 root root 10 Mar 18 13:06 virtio-pci-0000:04:00.0-part2 ->
../../vda2
```

Note: Pay very close attention to these details. Selecting the wrong disk may result in unmounting the wrong file system, which can have disastrous consequences.

4. Stop Apache and MySQL running on rdbk86r1 (x86) host:

```
service httpd stop
service mysqld stop
```

5. Unmount the disks that contain the file systems that will be moved to rdbkpmr1:

```
umount /srv/www
umount /var/lib/mysql
```

6. For completeness, log out of the LUN of the iSCSI target connected from rdbk86r1:

```
iscsiadm --mode=node --portal=129.40.23.230
--targetname=iqn.2024-03.rdbk86r1.pbm.ihost.com:lun1 --logout
```

7. Remove the records referring to the /srv/www and /var/lib/mysql from the rdbk86r1 host's /etc/fstab. (You may choose to retain the information before removing it; it is likely that you will use the exact same information on the new host, rdbkpmr1.)

Move iSCSI disks to Linux on IBM Z

1. Use SSH to connect again to the console of guest, rdbkpmr1. (No more work will be done using the console on the x86 host rdbk86r1.)

2. Discover the remote disk services that are running on the remote iSCSI target by using the following command:

```
iscsiadm --mode=discovery --type=sendtargets --portal=129.40.23.230
```

Example 7-42 shows three of the LUNs that are available from the iSCSI target. Our example uses only two of the LUNs.

Example 7-42 The LUNs discovered from the iSCSI target

```
[lnxadmin@rdbkpmr1 www]$ sudo iscsiadm -m discovery -t sendtargets -p  
129.40.23.230 129.40.23.230:3260,1 iqn.2024-03.rdbk86r1.pbm.ihost.com:lun1
```

Note that the output seen here is quite similar to the information seen on rdbk86r1, representing the iSCSI data that you gathered in the preceding steps, such as the IP address of the iSCSI target, the iSCSI Target Identifier, and the LUNs that the iSCSI Target is exporting. In this example, the correct iSCSI devices that will be mounted on rdbkpmr1 is LUN1 (which contains the Apache www file system).

3. Log in to the remote iSCSI disk, specifying the wanted LUN (LUN1) for the Apache www disk by using the following command:

```
iscsiadm --mode=node --portal=129.40.23.230  
--targetname=iqn.2024-03.rdbk86r1.pbm.ihost.com:lun1 --login
```

The output shown in Example 7-43 demonstrates a successful login.

Example 7-43 Successful login to iSCSI target's LUN1

```
[lnxadmin@rdbkpmr1 www]$ iscsiadm --mode=node --portal=129.40.23.230  
--targetname=iqn.2024-03.rdbk86r1.pbm.ihost.com:lun1 -- login  
Logging in to [iface: default, target: iqn.2024-03.rdbk86r1.pbm.ihost.com, portal:  
129.40.23.230,3260] (multiple)  
Login o [iface: default, target: iqn.2024-03.rdbk86r1.pbm.ihost.com, portal:  
129.40.23.230,3260] successful.
```

In your environment, it is highly likely that the iSCSI target requires more sophisticated authentication for login. In this simplistic example, the iSCSI target requires no credentials of any kind.

Note: The **--login** subcommand command on SUSE Enterprise Linux Server will cause the iSCSI initiator to connect only to the specified LUN, and consequently only the specified LUN will be viewable. However, with RHEL, the **--login** subcommand to the iSCSI target will allow you to see and manipulate all the LUNs that are available on the iSCSI target that are authorized by the login. This behavior on RHEL has one helpful consequence, but also a few challenges. The helpful consequence is that only one login step is needed. One particular challenge is that once the login is accomplished, all the iSCSI LUNs are given /dev/sdX assignments, whether they're all wanted.

4. See that the remote LUN is now connected to host rdbkpmr1:

```
ls -l /dev/disk/by-path/
```

Example 7-44 shows the partitions sda being mapped to iSCSI LUN1.

Example 7-44 iSCSI LUNs mapped to disk devices after login

```
[root@rdbkpmr1 ~]# ls -l /dev/disk/by-path  
total 0  
lrwxrwxrwx. 1 root root 11 Mar 22 11:39 ccw-0.0.0100 -> ../../dasda  
lrwxrwxrwx. 1 root root 12 Mar 22 11:39 ccw-0.0.0100-part1 -> ../../dasda1
```

```

lrwxrwxrwx. 1 root root 12 Mar 22 11:39 ccw-0.0.0100-part2 -> ../../dasda2
lrwxrwxrwx. 1 root root 11 Mar 22 11:39 ccw-0.0.0300 -> ../../dasdb
lrwxrwxrwx. 1 root root 12 Mar 22 11:39 ccw-0.0.0300-part1 -> ../../dasdb1
lrwxrwxrwx. 1 root root 11 Mar 22 11:39 ccw-0.0.0301 -> ../../dasdc
lrwxrwxrwx. 1 root root 12 Mar 22 11:39 ccw-0.0.0301-part1 -> ../../dasdc1
lrwxrwxrwx. 1 root root  9 Mar 25 16:41
ip-129.40.23.230:3260-iscsi-iqn.2024-03.rdbk86r1.pbm.ihost.com:lun1-lun-1 ->
../../sda

```

5. Add entries to `/etc/fstab` for the remote iSCSI disks to be routinely mounted in their appropriate places. Again, be sure to use the by-path designation. Example 7-45 shows a snippet from `/etc/fstab`, with the two new file system entries.

Example 7-45 New `/etc/fstab` containing the new MySQL and www disks on `rdbkpmr1`

```

# external iSCSI disk LUN1 for www filesystem
/dev/disk/by-path/ip-129.40.23.230:3260-iscsi-iqn.2024-03.rdbk86r1.pbm.ihost.com:lun1-lun-1
/var/www          xfs          nofail          1 2

```

6. Mount the remote iSCSI disks in their proper places using the following command:

```
mount /dev/sda /var/MediaWiki
```

The file systems from the remote iSCSI disks are now mounted and available on the new `rdbkpmr1` host.

Complete migration of services

The remaining tasks are critical and the most specialized. It is necessary to copy the Apache and MySQL configuration files from the x86 host to the Linux on IBM Z guest, then adapt them appropriately. Notice that the configuration of MediaWiki requires nothing special, since it was all moved when the www partition was moved. In some cases, the configurations may be simple enough to warrant a basic copy of the files. Other circumstances may be more complex, which will require rewriting the configuration files. But with the data having been migrated so effortlessly via iSCSI, it is easy to tolerate a small amount of editing of the configuration files.

For this example, a simple copy of the configuration files is all that is necessary. To do this:

1. Start an SSH console on `rdbkpmr1` (Linux on IBM Z).
2. Synchronize the Apache configuration files to `rdbkpmr1` from `rdbk86r1`. Use the method that makes the most sense for your environment, following this example:

```
rsync -qa rdbk86r1:/etc/apache2/* /etc/apache2/
rsync -qa rdbk86r1:/etc/my.cnf /etc/
```

3. Start the Apache and MySQL services on `rdbkpmr1`:

```
service httpd start
service mysqld start
```

4. Ensure that Apache and MySQL will start each time that the server is booted:

```
chkconfig httpd on
chkconfig mysqld on
```

Having successfully migrated Apache, MySQL, and their respective data, including the MediaWiki data, the MediaWiki application should now be functional.

Opening the MediaWiki URL using a browser, the web page will look similar to that shown in Figure 7-3, representing a successful installation of MediaWiki.

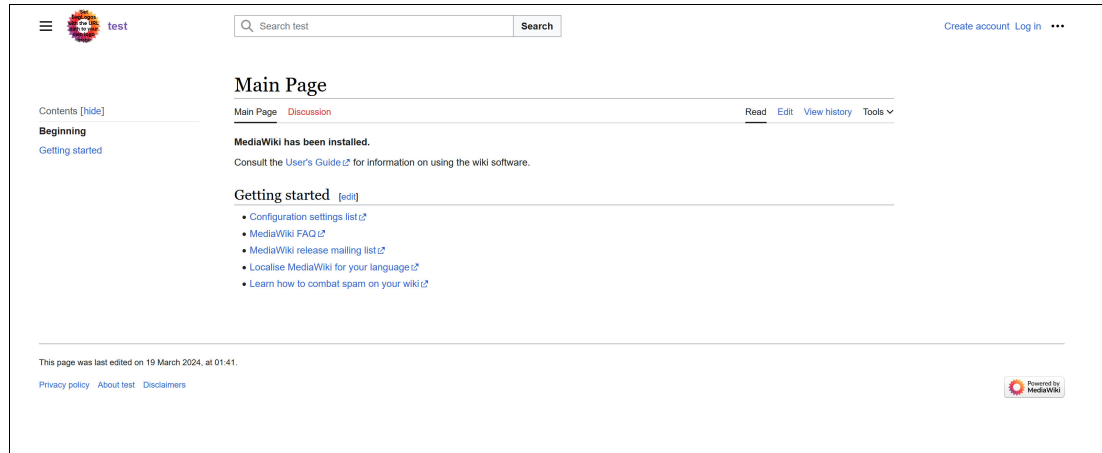


Figure 7-3 A successful migration of MediaWiki

7.6 Deploying OpenLDAP

Enterprises of all sizes need to manage the users of their computing resources. And with the user management comes the various characteristics of the user, such as user ID, authentication, file system rights, printer rights, and more, all needing to be managed. One of the most common products used for managing this data is the Lightweight Directory Access Protocol, commonly known as LDAP.

LDAP is widely used throughout the industry for directory services, as an open standard running over an IP network. Although there are several commercial LDAP products available, OpenLDAP is the implementation that is most commonly used in Linux. OpenLDAP is a fully featured suite of tools and applications. It is readily available as a workload on IBM Z from both RHEL and SUSE. LDAP is a perfect workload for Linux on IBM Z, due to the centrality of IBM Z among many other systems and services, its fast I/O, and its low CPU and memory overhead. And of course OpenLDAP is open source. Migrating OpenLDAP to Linux on IBM Z is straight forward.

In section 7.5, “Deploying MediaWiki and MySQL” on page 165, we installed a LAMP server with MediaWiki, and iSCSI external storage was used to facilitate the migration. In this example, the LDAP database on an x86 server will be exported, the database will be transferred to a Linux guest running on IBM Z, and the data will be imported into the LDAP service.

7.6.1 Analysis and planning

As with the MediaWiki example described in section 7.5, “Deploying MediaWiki and MySQL” on page 165, it is important that you follow Chapter 5, “Migration planning” on page 61, and Chapter 6, “Migration analysis” on page 71, knowing that appropriate planning and analysis should be performed before any migration activity. The checklists have been created to help identify the many considerations that should be made which will help prevent problems migrating.

Again for this example scenario, the z/VM guest has already been set up and a minimal Linux operating system has been installed.

The Linux guest is called LNSUDB2 and is running SUSE Enterprise Linux Server11 SP3, with one virtual CPU and 1 GB of virtual memory. An OpenLDAP server typically does not require a large amount of CPU or RAM running on Linux on IBM Z. It is presumed that an adequate RPM repository installation source is already set up and available for the installation of the application software that will be used.

The x86 server is called zs4p01-r1 and is running RHEL 6.4. For this example, this is the current OpenLDAP server providing directory services for the hypothetical organization. This server has a very rudimentary (small) LDAP directory already configured.

Although there is much to consider when setting up an enterprise directory service, a very simple [OpenLDAP](#) scenario will be covered here.

This example is a stand-alone server with a local, non-replicated directory service. Nevertheless, migrating an existing OpenLDAP installation on x86 to Linux on IBM Z should be very straight forward.

7.6.2 Installing LDAP software

The OpenLDAP server is technically a very simple application, consisting of a single package. Consequently installing the software is relatively easy. The software must first be installed on the Linux on IBM Z guest before other migration steps should be attempted. If you are going to install OpenLDAP on SUSE Enterprise Linux Server, run the following command to install the package on SUSE Enterprise Linux Server:

```
zypper install openldap2
```

To install OpenLDAP on RHEL, run the following command:

```
yum install openldap-servers
```

7.6.3 Configuring the OpenLDAP service

The principal player in the OpenLDAP server suite of applications is the Standalone LDAP Daemon, known as **slapd**. This example configures the **slapd** service to operate as a stand-alone, local, non-replicated directory. The package, in RPM format, contains functional sample configuration files, which will serve as the basis of the example service that is configured here.

The initial configuration of OpenLDAP on SUSE Enterprise Linux Server running on IBM Z will be accomplished using YaST, while a parallel example on Red Hat will be done by manually modifying configuration files and running commands.

Before migrating the LDAP database to Linux on IBM Z, it is necessary to establish a basic configuration of OpenLDAP. Using different terminology, the OpenLDAP configuration must be started, also known as bootstrapped.

Note: OpenLDAP maintains its configuration using one of two different configuration methods. The “old” method involves maintaining the primary configuration in `/etc/openldap/slapd.conf`. This method is very simple, but does not have as many features. The “new” way (called the `cn=config` format) uses several configuration files below `/etc/openldap/slapd.d/`. The default behavior with OpenLDAP 2.4 is to use the `cn=config` method.

Configuring OpenLDAP on SUSE Enterprise Linux Server using YaST

All of the activities to create a basic configuration of OpenLDAP are facilitated by the LDAP server YaST module. By following a few simple screens in YaST, the LDAP services can be configured and running in short order. To do this, perform the following steps:

1. From a command prompt, start YaST, calling specifically the ldap-server module:

```
yast2 ldap-server
```

Figure 7-4 shows the first panel.

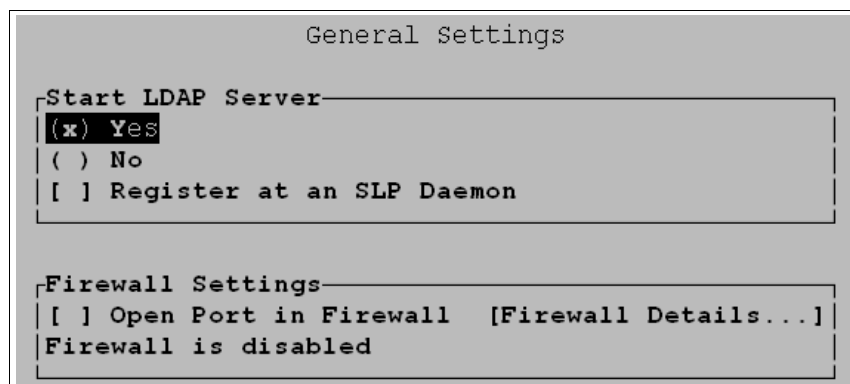


Figure 7-4 yast2 ldap-server module

Select “Yes” to start the LDAP server automatically, and be certain to open a port in the firewall. In our example, since the firewall is disabled, we did not have the option to select “Open Port in Firewall”.

Press F-10 to go to the next panel.

2. Select “Stand-alone server” as the server type, as shown in Figure 7-5.

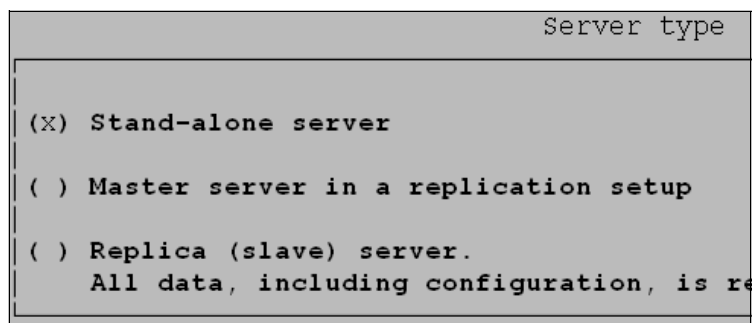


Figure 7-5 Stand-alone server type of the LDAP server

Press F-10 to go the next panel.

3. Select the proper security settings for OpenLDAP, as shown in Figure 7-6 on page 178.

Figure 7-6 TLS and SSL certificate settings for OpenLDAP

Using a proper SSL certificate is highly recommended, but not necessary for this demonstration. Here, we use the self-signed system certificates generated when SUSE Enterprise Linux Server was installed. More importantly, note that using SSL for the LDAP protocol (LDAPS) is essential. Without LDAPS, passwords and other sensitive data will be exchanged with the LDAP server in plaintext. No one needs or even should want that.

Press F-10 to go to the next panel.

- Figure 7-7 illustrates the **Basic Database Settings** panel, which includes fields for setting an administrative password for LDAP.

Figure 7-7 Basic database settings for OpenLDAP configuration

In a production environment, proper distinguished name (DN) data should be entered, but for this demonstration it is adequate to use the sample values supplied by YaST. What is most important to note here is the need to provide an administrator password. Best practices dictate that this should not be the same as the system's root password, and all other best practices for creating an administrative password should likewise be employed. For this demonstration, the password "ldapadmin" will be used.

Press F-10 to go to the next panel.

- The configuration summary is displayed, as shown in Figure 7-8 on page 179.

```

LDAP Server Configuration Summary

Startup Configuration

Start LDAP Server: Yes

Register at SLP Service: No

Create initial Database with the following:

Database Suffix: dc=itso,dc=ibm,dc=com

Administrator DN: cn=Administrator,dc=itso,dc=ibm,dc=com

```

Figure 7-8 OpenLDAP configuration summary

With all the configuration information sufficiently gathered, the YaST configuration steps can be completed, by pressing F-10 to finish. The configuration files are written, and the slapd daemon is started. The running daemon process can be seen in Example 7-46. Note that the **-F /etc/openldap/slapd.d** argument indicates that the service is configured using the **cn=config** feature format.

Example 7-46 slapd daemon shown running using the **cn=config** method

```

lnsldb2:- # ps -ef | grep slapd
ldap      17224      1      0 16:26 ?          00:00:00 /usr/lib/openldap/slapd -h ldap:
/// ldaps:/// ldapi:/// -F /etc/openldap/slapd.d -u ldap -g ldap -o slp=off
root      17251      7470      0 16:27 pts/0      00:00:00 grep slapd

```

Configuring OpenLDAP manually on Red Hat Enterprise Linux

The configuration on the Red Hat Enterprise Linux (RHEL) server is also a relatively easy task because all that is needed is a basic, bootstrappable configuration. This basic configuration by itself is not useful for running a proper directory, but it will allow the migration of the openLDAP directory from another server. OpenLDAP 2.4 on RHEL6 also uses the **cn=config** feature configuration format by default:

1. Ensure that the **slapd** daemon is running.

service slapd start

2. From a command prompt on the RHEL server, edit a basic OpenLDAP configuration file, perhaps using **vi** as in the following example:

vi /tmp/config.itso.ibm.com.ldif

Put the content shown in Example 7-47 into the file:

Example 7-47 /tmp/config.itso.ibm.com.ldif file to bootstrap the OpenLDAP database

```

dn: olcDatabase=hdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {1}hdb
olcSuffix: dc=itso,dc=ibm,dc=com
olcDbDirectory: /var/lib/ldap
olcRootDN: cn=Administrator,dc=itso,dc=ibm,dc=com
olcRootPW: ldapadmin
olcAccess: to attrs=userPassword by dn="cn=Administrator,dc=itso,dc=ibm,dc=com"
write by anonymous auth by self write by * none
olcAccess: to attrs=shadowLastChange by self write by * read

```

```
olcAccess: to dn.base="" by * read
olcAccess: to * by dn="cn=Administrator,dc=itso,dc=ibm,dc=com" write by * read
```

Save the file, and exit the editor.

3. Bootstrap the database and import the configuration from the file created in Example 7-47 on page 179 using the following command:

```
ldapadd -Y EXTERNAL -H ldapi:/// -f /tmp/config.itso.ibm.com.ldif
```

Now the basic configuration of OpenLDAP will allow a migration of the database.

7.6.4 Export OpenLDAP data from x86 server

The LDAP directory tree running on the x86 server now needs to be exported, so that the data can be transferred to the Linux guest on IBM Z. To do this, perform the following steps:

1. Connect to the x86 host, zs4p01-r1, using an SSH. We are doing this on an RHEL server.
2. Stop the slapd daemon so that the data can be exported from OpenLDAP:

```
service slapd stop
```

3. Export the data from the OpenLDAP database. The tool used to accomplish this is called slapcat, and this is a common method of extracting whole data sets from OpenLDAP. The output is written in LDAP Data Interchange Format (LDIF), which is a standard plain text data interchange format for representing LDAP:

```
slapcat -b 'dc=itso,dc=ibm,dc=com' -l /tmp/migrate.ldif
```

The “-l” argument tells slapcat to export the database (in the LDIF format) to the file /tmp/migrate.ldif. The “-b” argument identifies the specific domain of data to export (known as the suffix in the OpenLDAP vernacular).

4. (Optional) Restart the slapd daemon on zs4p01-r1. Since the daemon is being migrated to another server, it may not be necessary to restart it.

```
service slapd start
```

5. Transfer the database file to the Linux guest, LNSUDB2, running on IBM Z. Use the transfer mechanism that is most suitable. This example uses a utility software and network protocol called **rsync**:

```
rsync /tmp/migrate.ldif 9.12.7.90:/tmp/
```

Note that the server with the IP address 9.12.7.90 is LNSUDB2 and is the Linux guest on IBM Z. Provide appropriate credentials when prompted. When the transfer is complete, the process of exporting the data from this x86 server to the Linux guest running on IBM Z has been successfully completed.

7.6.5 Import OpenLDAP data to Linux on IBM Z

In the previous section, the OpenLDAP database export file was transferred to Insudb2, the Linux guest running on IBM Z. All that is required now is to import the data and start the OpenLDAP daemon:

1. Reconnect to the IBM Z guest, Insudb2, using SSH.
2. Ensure that slapd is not running. Importing data for migration requires that the service is not running:

```
service slapd stop
```

3. Import the data that was copied. This process employs a tool called slapadd. This is a common method of importing whole data sets into OpenLDAP:

```
slapadd -F /etc/openldap/slapd.d \
-b 'dc=itso,dc=ibm,dc=com' -l /tmp/migrate.ldif
```

Because the basic configuration was established in section 7.6.3, “Configuring the OpenLDAP service” on page 176, the itso.ibm.com domain already exists in the new OpenLDAP database, making it very easy to import the data. The “-b” argument identifies the domain, and the “-l” argument indicates the LDIF file from which the database information will be imported.

A successful import shows “100%” success, as illustrated in Example 7-48. Any value other than 100% means that something went wrong and the import of the data was not successful.

Example 7-48 Import of OpenLDAP data is 100% successful

```
lnsldb2:- # slapadd -F /etc/openldap/slapd.d -b 'dc=itso,dc=ibm,dc=com' -l /tmp/migrate.ldif
hdb_monitor_db_open: monitoring disabled; configure monitor database to enable
_##### 100.00% eta none elapsed none fast!
Closing DB...
```

4. Once the database has been successfully imported, OpenLDAP can be started again, ready to receive queries:

```
service slapd start
```

7.6.6 Verify that OpenLDAP is working

The slapd process is running, and sample data is presumed to exist in the directory, but that does not necessarily mean that OpenLDAP is usable by any clients. It is important to test that the LDAP server responds to client requests. In this example, the user “fred” is queried:

```
ldapsearch -xLLL -H ldapi:/// -b "dc=itso,dc=ibm,dc=com" uid=fred sn givenName cn
```

Example 7-49 shows the results of the ldapsearch query.

Example 7-49 Output from ldapsearch, showing user fred exists in the directory

```
lnsldb2:- # ldapsearch -xLLL -H ldapi:/// -b "dc=itso,dc=ibm,dc=com" uid=fred sn
givenName cn
dn: uid=fred,ou=employees,dc=itso,dc=ibm,dc=com
sn: frandsen
cn: fred
```

But in the preceding example, the OpenLDAP client and the server are both running on the same system, LNSUDB2. That is not necessarily a convincing demonstration. A better verification is whether an external client can query the OpenLDAP server over the network. Example 7-50 shows that a different client, zs4p01-s1, queries the LDAP directory running on lnsldb2 (9.12.7.90).

Example 7-50 Output from ldapsearch, querying the LDAP directory over the network

```
zs4p01-s1:~ # ldapsearch -xLLL -H ldap://9.12.7.90 \
> -b "dc=itso,dc=ibm,dc=com" \
> uid=fred sn givenName cn
dn: uid=fred,ou=employees,dc=itso,dc=ibm,dc=com
```

```
sn: frandsen
cn: fred
```

This second verification in Example 7-50 on page 181 indicates a successful migration of an OpenLDAP service from Linux on x86 to Linux on IBM Z. Not only that, but the service has been quite easily migrated from a system running RHEL to one running SUSE Enterprise Linux Server. OpenLDAP, Linux, and IBM Z are all very happy regardless of the distribution, and the migration of OpenLDAP is unhampered regardless of the distribution.

7.7 Deploying central log server

As you saw in section 6.7.10, “Logging and recording events” on page 121, forwarding local log records to a remote secure system is a good practice to keep your log records safe. When someone does attempt to attack one of the servers, they will probably try to clean up their tracks. By using remote centralized log servers, you can keep a safe copy even if they remove the local copies or stop the service. Also, you will be able to centralize all logs from your environment and use a real-time search and analytics tool to create business insights or a monitoring tool.

To create a centralized log server we will use the default log daemon from Red Hat Enterprise Linux, **rsyslog** (version 8). You can verify the rsyslog version by issuing command listed in Example 7-51.

Example 7-51 Command to verify the rsyslog version with output

```
# rsyslogd -v
rsyslogd 8.2102.0-117.el9 (aka 2021.02) compiled with:
  PLATFORM: s390x-ibm-linux-gnu
  PLATFORM (lsb_release -d):
  FEATURE_REGEX:Yes
  GSSAPI Kerberos 5 support:Yes
  FEATURE_DEBUG (debug build, slow code):No
  32bit Atomic operations supported:Yes
  64bit Atomic operations supported:Yes
  memory allocator:system default
  Runtime Instrumentation (slow code):No
  uuid support:Yes
  systemd support:Yes
  Config file:/etc/rsyslog.conf
  PID file: /var/run/rsyslogd.pid
  Number of Bits in RainerScript integers: 64
```

See <https://www.rsyslog.com> for more information.

7.7.1 Analysis and planning

Use the logical volume manager (LVM) to create a logical volume for log files because log files tend to grow very fast, and with different hosts writing logs to a centralized log server at the same time, log files can fill your disk even faster. So, leave some space available in a volume group that can be used in case of emergency. Another important advantage using LVM is the flexibility it will bring to manage the file system which stores the log files.

7.7.2 Initial configuration

The default path for the **rsyslog** configuration file is `/etc/rsyslog.conf`. It is used to control the output of the `syslogd` daemon log files, which contains key words that define the message route and global options. You can see all the available global options by issuing the command:

```
man rsyslog.conf
```

The `rsyslogd` daemon reads the configuration file when it is activated.

For more information about `rsyslog`, please access the official [rsyslog documentation](#).

To use remote logging, we need to configure the server and the client.

Our lab information

Rsyslog server = `rdbkpmr1.pbm.ihost.com`

Rsyslog client = `rdbk86r1.pbm.ihost.com`

Server configuration

Use the steps in this section to enable a Linux server to receive logs from any number of remote clients. For this publication, we will use a TCP 30514 port to set up a syslog server on `rdbkpmr1.pbm.ihost.com`.

1. Check current firewall rules as shown in Example 7-52.

Example 7-52 Command to check firewall rules

```
# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enc640
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 21/tcp 5901/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Note that, TCP 30514 port is not listed in the ports line, so our next step is to open the local firewall to allow this communication.

2. Configure `firewalld` to allow incoming `rsyslog` traffic on port 30514, as shown in Example 7-53.

Example 7-53 Command to open port

```
firewall-cmd --permanent --add-port=30514/tcp
firewall-cmd --reload
```

3. To confirm port 30514 is now allowed to communicate, issue the command shown in Example 7-54. Note that our port is now listed (highlighted in **red** in our example).

Example 7-54 Verify port is now open for communication

```
# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enc640
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 21/tcp 5901/tcp 30514/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

4. List the SELinux ports by entering the command shown in Example 7-55.

Example 7-55 List SELinux ports

```
# semanage port -l | grep syslog
syslog_tls_port_t      tcp      6514, 10514
syslog_tls_port_t      udp      6514, 10514
syslogd_port_t         tcp      601, 20514
syslogd_port_t         udp      514, 601, 20514
```

5. Add the port, TCP 30514, to syslogd_port_t SELinux by using the command shown in Example 7-56.

Example 7-56 Add port to syslogd_port_t SELinux

```
semanage port -a -t syslogd_port_t -p tcp 30514
```

6. Review the SELINUX ports to ensure port was added, as shown in Example 7-57.

Example 7-57 Command to review ports

```
# semanage port -l | grep syslog
syslog_tls_port_t      tcp      6514, 10514
syslog_tls_port_t      udp      6514, 10514
syslogd_port_t         tcp      30514, 601, 20514
syslogd_port_t         udp      514, 601, 20514
```

7. Create a file called logserver.conf in the /etc/rsyslog.d/ directory and insert the content shown in Example 7-58. This is a custom configuration file. This way, during Linux Patches or Upgrades, if the Rsyslog package gets updated, the main file (/etc/rsyslog.conf) can be replaced with a new version from vendor and you retain your custom settings.

Example 7-58 Create a custom configuration file

```
# Define templates before the rules that use them
# Per-Host templates for remote systems
template(name="TplAuthpriv" type="list") {
  constant(value="/var/log/remote/auth/")
  property(name="hostname")
  constant(value="/")
  property(name="programname" SecurePath="replace")
}
```

```

        constant(value=".log")
    }

    template(name="TplMsg" type="list") {
        constant(value="/var/log/remote/msg/")
        property(name="hostname")
        constant(value="/")
        property(name="programname" SecurePath="replace")
        constant(value=".log")
    }

    # Provides TCP syslog reception
    module(load="imtcp")

    # Adding this ruleset to process remote messages
    ruleset(name="remote1"){
        authpriv.*    action(type="omfile" DynaFile="TplAuthpriv")
        *.info;mail.none;authpriv.none;cron.none
        action(type="omfile" DynaFile="TplMsg")
    }

    input(type="imtcp" port="30514" ruleset="remote1")

```

8. Test the syntax of the `/etc/rsyslog.conf` file by using the command shown in Example 7-59.

Example 7-59 Test syntax

```

# rsyslogd -N 1
rsyslogd: version 8.2102.0-117.e19, config validation run (level 1), master
config /etc/rsyslog.conf
rsyslogd: End of config validation run. Bye.

```

9. Ensure that the `rsyslog` service is running and enabled on the logging server by using the command shown in Example 7-60.

Example 7-60 Verify rsyslog service is running

```

systemctl status rsyslog
? rsyslog.service - System Logging Service
   Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled; preset:
   enabled)
   Active: active (running) since Wed 2024-01-24 16:25:27 EST; 1 month 17
   days ago
     Docs: man:rsyslogd(8)
           https://www.rsyslog.com/doc/
    Main PID: 1281 (rsyslogd)
      Tasks: 3 (limit: 100835)
     Memory: 5.0M
        CPU: 39.568s
    CGroup: /system.slice/rsyslog.service
            ??1281 /usr/sbin/rsyslogd -n

Feb 11 00:10:23 rdbkpmr1.pbm.ihost.com rsyslogd[1281]: [origin
software="rsyslogd" swVersion="8.2102.0-117.e19" x-pid="1281"
x-info="https://www.rsyslog.com"] rsyslogd was HUPed
Feb 18 00:00:23 rdbkpmr1.pbm.ihost.com systemd[1]: rsyslog.service: Sent signal
SIGHUP to main process 1281 (rsyslogd) on client request.

```

```
Feb 18 00:10:23 rdbkpmr1.pbm.ihost.com rsyslogd[1281]: [origin
software="rsyslogd" swVersion="8.2102.0-117.e19" x-pid="1281"
x-info="https://www.rsyslog.com"] rsyslogd was HUPed
Feb 24 03:01:01 rdbkpmr1.pbm.ihost.com rsyslogd[1281]: imjournal: journal files
changed, reloading... [v8.2102.0-117.e19 try https://www.rsyslog.com/e/0 ]
Feb 25 00:00:23 rdbkpmr1.pbm.ihost.com systemd[1]: rsyslog.service: Sent signal
SIGHUP to main process 1281 (rsyslogd) on client request.
Feb 25 00:10:23 rdbkpmr1.pbm.ihost.com rsyslogd[1281]: [origin
software="rsyslogd" swVersion="8.2102.0-117.e19" x-pid="1281"
x-info="https://www.rsyslog.com"] rsyslogd was HUPed
Mar 03 00:00:23 rdbkpmr1.pbm.ihost.com systemd[1]: rsyslog.service: Sent signal
SIGHUP to main process 1281 (rsyslogd) on client request.
Mar 03 00:10:23 rdbkpmr1.pbm.ihost.com rsyslogd[1281]: [origin
software="rsyslogd" swVersion="8.2102.0-117.e19" x-pid="1281"
x-info="https://www.rsyslog.com"] rsyslogd was HUPed
Mar 10 00:00:23 rdbkpmr1.pbm.ihost.com systemd[1]: rsyslog.service: Sent signal
SIGHUP to main process 1281 (rsyslogd) on client request.
Mar 10 00:10:23 rdbkpmr1.pbm.ihost.com rsyslogd[1281]: [origin
software="rsyslogd" swVersion="8.2102.0-117.e19" x-pid="1281"
x-info="https://www.rsyslog.com"] rsyslogd was HUPed
```

10. Restart the rsyslog service by using the following command:

```
# systemctl restart rsyslog
```

11. Optional: If rsyslog is not enabled, ensure the rsyslog service starts automatically after reboot by using the following command:

```
# systemctl enable rsyslog
```

12. Review the listening ports to ensure port is opened. Example 7-61 shows the command and its output.

Example 7-61 Review the listening ports to ensure port is opened.

```
# netstat -tnlp | grep rsyslog
tcp        0      0 0.0.0.0:30514          0.0.0.0:*             LISTEN
210743/rsyslogd
tcp6       0      0 :::30514              :::*                   LISTEN
210743/rsyslogd
```

or

```
# ss -tlnp | grep "rsyslog"
tcp        LISTEN 0      25          0.0.0.0:30514      0.0.0.0:*
users:(("rsyslogd",pid=210743,fd=4))
tcp        LISTEN 0      25          [::]:30514        [::]:*
users:(("rsyslogd",pid=210743,fd=5))
```

As shown in Example 7-61, the port is opened to external network connections.

The configuration of a log server is now complete and ready to receive and store log files from the other systems in your environment. In the next section, we provide the steps to set up the client side.

Client Configuration

In this section, we provide steps that will instruct our Rsyslog client to forward all its logs to the central Rsyslog server called rdbkpmr1.pbm.ihost.com in our lab environment.

1. Create a new file in the /etc/rsyslog.d/ directory named 10-default.conf, and insert the content shown in Example 7-62.

Example 7-62 Client custom configuration file

```
*.* action(type="omfwd"
    queue.type="linkedlist"
    queue.filename="remote_fwd"
    action.resumeRetryCount="-1"
    queue.saveOnShutdown="on"
    target="rdbkpmr1.pbm.ihost.com" port="30514" protocol="tcp"
)
```

Where:

The queue.type="linkedlist" setting enables a LinkedList in-memory queue,

The queue.filename setting defines a disk storage. The backup files are created with the remote_fwd prefix in the working directory specified by the preceding global workDirectory directive.

The action.resumeRetryCount -1 setting prevents rsyslog from dropping messages when retrying to connect if server is not responding,

The queue.saveOnShutdown="on" setting saves in-memory data if rsyslog shuts down.

The last line forwards all received messages to our logging server.

With this configuration, rsyslog sends messages to the server but keeps messages in memory if the remote server is not reachable. A file on disk is created only if rsyslog runs out of the configured memory queue space or needs to shut down, which benefits the system performance.

2. Test the syntax of the /etc/rsyslog.conf file by using the command shown in Example 7-63.

Example 7-63 Verify syntax

```
# rsyslogd -N 1
rsyslogd: version 8.2102.0-117.el9, config validation run (level 1), master
config /etc/rsyslog.conf
rsyslogd: End of config validation run.
Bye
```

3. Review the SELinux ports by entering the command shown in Example 7-64.

Example 7-64 Review SELinux ports

```
# semanage port -l | grep syslog
syslog_tls_port_t      tcp      6514, 10514
syslog_tls_port_t      udp      6514, 10514
syslogd_port_t         tcp      601, 20514
syslogd_port_t         udp      514, 601, 20514
```

4. Add the TCP 10514 port to syslogd_port_t SELinux type by using the following command:
semanage port -a -t syslogd_port_t -p tcp 30514
5. Review the SELINUX ports to ensure port was added, as shown in Example 7-65.

Example 7-65 Verify new port has been added

```
# semanage port -l | grep syslog
syslog_tls_port_t          tcp      6514, 10514
syslog_tls_port_t          udp      6514, 10514
syslogd_port_t             tcp      30514, 601, 20514
syslogd_port_t             udp      514, 601,
20514
```

6. Restart the rsyslog service by using the following command:

```
# systemctl restart rsyslog
```

If you don't update SELINUX as detailed above, you may see the errors shown in Example 7-66 in the /var/log/messages file.

Example 7-66 Sample error messages

```
Mar 12 12:57:15 rdbk86r1 rsyslogd[2995]: cannot connect to
rdbkpmr1.pbm.ihost.com:30514: Permission denied [v8.2102.0-117.e19 try
https://www.rsyslog.com/e/2027 ]
Mar 12 12:57:15 rdbk86r1 rsyslogd[2995]: action 'action-0-builtin:omfwd' suspended
(module 'builtin:omfwd'), retry 0. There should be messages before this one giving
the reason for suspension. [v8.2102.0-117.e19 try https://www.rsyslog.com/e/2007 ]
Mar 12 12:57:15 rdbk86r1 rsyslogd[2995]: cannot connect to
rdbkpmr1.pbm.ihost.com:30514: Permission denied [v8.2102.0-117.e19 try
https://www.rsyslog.com/e/2027 ]
Mar 12 12:57:45 rdbk86r1 rsyslogd[2995]: cannot connect to
rdbkpmr1.pbm.ihost.com:30514: Permission denied [v8.2102.0-117.e19 try
https://www.rsyslog.com/e/2027 ]
Mar 12 12:58:15 rdbk86r1 rsyslogd[2995]: cannot connect to
rdbkpmr1.pbm.ihost.com:30514: Permission denied [v8.2102.0-117.e19 try
https://www.rsyslog.com/e/2027 ]
```

If you get these errors, repeat the SELINUX commands listed in steps 2-6. Otherwise, rsyslog won't be permitted to send logs to TCP 30514 port.

Testing rsyslog configuration

The following steps can be used to verify that the client system sends messages to the server.

1. On the client system, send a test message by using the following command:

```
logger "My LOG test"
```

You can issue the `tail /var/log/messages` command to confirm the test message was registered.

2. While connected to the rsyslog server (in our case, rdbkpmr1.pbm.ihost.com), issue the command shown in Example 7-67 on page 188. Verify that a folder with the client system name was created under /var/log/remote/msg, as shown in Example 7-67 on page 188.

Example 7-67 Verify client system is available

```
# ls -la /var/log/remote/msg/
total 0
drwx----- 3 root root 22 Mar 12 14:08 .
drwx----- 3 root root 17 Mar 12 14:08 ..
```

```
drwx-----. 2 root root 109 Mar 12 14:08 rdbk86r1
```

3. If it exists, open the log file found in the client system folder by using the command shown in Example 7-68.

Example 7-68 Open the file

```
# cat /var/log/remote/msg/rdbk86r1/root.log
Mar 12 12:58:32 rdbk86r1 root[3002]: My LOG test
Mar 12 13:02:53 rdbk86r1 root[3015]: My LOG test
Mar 12 13:09:06 rdbk86r1 root[12855]: My LOG test
```

Note: The log contains the user name of the user that entered the logger command, in this case root.

You can review other log files created by rsyslog

```
# ls -la /var/log/remote/msg/rdbk86r1/
total 24
drwx-----. 2 root root 109 Mar 12 14:08 .
drwx-----. 3 root root 22 Mar 12 14:08 ..
-rw-----. 1 root root 97 Mar 12 14:08 dbus-broker-launch.log
-rw-----. 1 root root 658 Mar 12 14:08 kernel.log
-rw-----. 1 root root 148 Mar 12 14:09 root.log
-rw-----. 1 root root 5244 Mar 12 14:08 rsyslogd.log
-rw-----. 1 root root 1414 Mar 12 14:08 systemd.log
```

For more information, see the official [Red Hat documentation](#).

7.7.3 Migrating with syslog-ng

Rsyslog can be used as a tool for your migration. You can set up a centralized log server to keep a copy of the log files for all the servers that you are migrating. Therefore, if a problem happens on the server and you lose access, you can easily fetch information or error messages.

If you are migrating an existing rsyslog server/client, you need to check if rsyslog is installed on the target server and ensure that the former configuration file is compatible to the version available on Linux on IBM Z. To migrate the old data, you can use an LVM snapshot to transfer the logical volume to the new server. Other commands such as **tar** and **rsync** can be used to transfer the old log files. You can see a practical example of LVM snapshot, **tar**, and **rsync** in the IBM Redbooks publication, *Set up Linux on IBM System z for Production*, SG24-8137.

7.7.4 Viewing logs: Grafana Loki

Once you have a central log server set up, making the logs viewable and searchable might be useful to help the teams performing the migrations. It is certainly possible for staff to log on to the log server directly and use tools like **grep** and **less** to view log files, but more complex log viewing tasks such as viewing different log data simultaneously (chronyd and smbd, for example) requires a reasonable amount of command-line skill.

Grafana is an open-source project that produces a web-based graphical tool for visualizing system data.

Grafana is usually associated with system performance metrics, but another part of the Grafana project is a tool called Loki which can manage log data. Grafana Loki presents a fairly low-overhead system for visualizing log data. Installing Grafana Loki onto your centralized log server could be an ideal way to present your log data.

Grafana Loki architecture

A component view of a Grafana Loki installation would look like Figure 7-9.

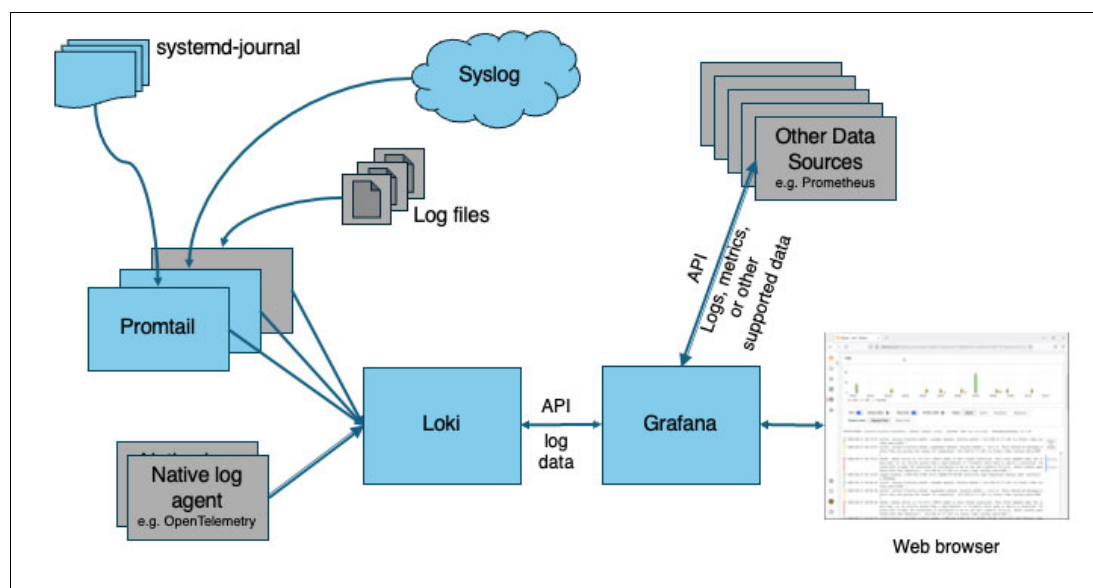


Figure 7-9 Grafana architecture, focusing on Loki

In our lab environment, we implemented a subset of what Grafana Loki is capable of. In Figure 7-9, the blue boxes show the capabilities and components we implemented. The gray boxes show examples of other capabilities that can be supported by Loki and Grafana in general (and there are still features and capabilities not shown).

Our Loki implementation is comprised of three main components:

Grafana	The main presentation layer, which draws from the configured data sources to display system information.
Loki	The data source that manages log data, managing labels for searching, and responding to queries from Grafana.
Promtail	Pre-processor of log data: parses raw log data into fields, creating labels for Loki to manage, and optionally transforming log data.

Promtail receives the raw log data and performs optional processing of the data to organize it into data fields and adds labels for indexing. Loki stores and organizes the log data and maintains indexes based on the labels set by Promtail. Finally, Grafana queries Loki for the available labels and reads message data from Loki for presentation.

Log data labels

The key data construct in Loki is the label, which is a named field that can be used by Loki (and Grafana) to sort and organize your log data.

One of the reasons Loki has lower resource overhead than other solutions is that it does not do full-text indexing of log entries. Loki uses the labels of the log messages to allow high-level searching and sorting of messages.

This means it is very important to use only the most meaningful and useful labels in your Promtail configuration: if too many fields are turned into labels, the overhead of Loki will grow and performance may decline.

Note: More information about Grafana Loki and its operation can be found via the Grafana Loki web site at <https://grafana.com/oss/loki/>.

Building and deploying Loki for Linux on IBM Z

At the time of writing, there are no Red Hat-supplied packages for Loki. It is easy to build from source, however. You may also be able to engage a third-party to build, supply, and support Loki for you.

In our environment we built Loki from the upstream source code. To build Grafana Loki, the packages `golang-bin` and `systemd-devel` must be installed first. Example 7-69 shows the steps we took and the commands we used (after the two prerequisite packages were installed).

Example 7-69 Building Grafana Loki from source

```
[lnxadmin@rdbkpmr1 ~]$ git clone https://github.com/grafana/loki.git ❶
Cloning into 'loki'...
remote: Enumerating objects: 156680, done.
remote: Counting objects: 100% (2822/2822), done.
remote: Compressing objects: 100% (1795/1795), done.
remote: Total 156680 (delta 1241), reused 2329 (delta 905), pack-reused 153858
Receiving objects: 100% (156680/156680), 271.03 MiB | 57.58 MiB/s, done.
Resolving deltas: 100% (103334/103334), done.
[lnxadmin@rdbkpmr1 ~]$ cd loki ❷
[lnxadmin@rdbkpmr1 loki]$ git switch use_go_120_6 ❸
branch 'use_go_120_6' set up to track 'origin/use_go_120_6'.
Switched to a new branch 'use_go_120_6'
[lnxadmin@rdbkpmr1 loki]$ make all ❹
CGO_ENABLED=1 go build -ldflags "-s -w -X
github.com/grafana/loki/pkg/util/build.Branch=use_go_120_6 -X
github.com/grafana/loki/pkg/util/build.Version=use_go_120_6-8d26909 -X
github.com/grafana/loki/pkg/util/build.Revision=8d2690975 -X
github.com/grafana/loki/pkg/util/build.BuildUser=lnxadmin@rdbkpmr1.pbm.ihost.com -X
github.com/grafana/loki/pkg/util/build.BuildDate=2024-03-20T05:07:43Z" -tags netgo -o
clients/cmd/promtail/promtail ./clients/cmd/promtail
CGO_ENABLED=0 go build -ldflags "-extldflags \"-static\" -s -w -X
github.com/grafana/loki/pkg/util/build.Branch=use_go_120_6 -X
github.com/grafana/loki/pkg/util/build.Version=use_go_120_6-8d26909 -X
github.com/grafana/loki/pkg/util/build.Revision=8d2690975 -X
github.com/grafana/loki/pkg/util/build.BuildUser=lnxadmin@rdbkpmr1.pbm.ihost.com -X
github.com/grafana/loki/pkg/util/build.BuildDate=2024-03-20T05:07:43Z" -tags netgo -o
cmd/logcli/logcli ./cmd/logcli
CGO_ENABLED=0 go build -ldflags "-extldflags \"-static\" -s -w -X
github.com/grafana/loki/pkg/util/build.Branch=use_go_120_6 -X
github.com/grafana/loki/pkg/util/build.Version=use_go_120_6-8d26909 -X
github.com/grafana/loki/pkg/util/build.Revision=8d2690975 -X
github.com/grafana/loki/pkg/util/build.BuildUser=lnxadmin@rdbkpmr1.pbm.ihost.com -X
github.com/grafana/loki/pkg/util/build.BuildDate=2024-03-20T05:07:43Z" -tags netgo -o
cmd/loki/loki ./cmd/loki
CGO_ENABLED=0 go build -ldflags "-extldflags \"-static\" -s -w -X
github.com/grafana/loki/pkg/util/build.Branch=use_go_120_6 -X
github.com/grafana/loki/pkg/util/build.Version=use_go_120_6-8d26909 -X
github.com/grafana/loki/pkg/util/build.Revision=8d2690975 -X
github.com/grafana/loki/pkg/util/build.BuildUser=lnxadmin@rdbkpmr1.pbm.ihost.com -X
```

```
github.com/grafana/loki/pkg/util/build.BuildDate=2024-03-20T05:07:43Z" -tags netgo -o
cmd/loki-canary/loki-canary ./cmd/loki-canary
[lnxadmin@rdbkpmr1 loki]$ sudo cp ./clients/cmd/promtail/promtail ./cmd/logcli/logcli
./cmd/loki/loki ./cmd/loki-canary/loki-canary /usr/local/bin/ 5
[lnxadmin@rdbkpmr1 loki]$ sudo chmod +x /usr/local/bin/{promtail,logcli,loki,loki-canary} 6
[lnxadmin@rdbkpmr1 loki]$
```

The build actions shown are:

1. The Grafana Loki repository was cloned from GitHub.
2. We changed into the cloned directory.
3. The version of Go distributed with RHEL is older than the version used for the mainline Loki development. We switched to a branch in the Loki code which will work with the version of Go shipped with RHEL.
4. We issued the “make” command to build the Loki binaries.

Once Step 4 was complete, the Loki binaries were built and ready to run and could have been executed from that path. For convenience we copied them from the build location to /usr/local/bin, as shown in Step 5. We also made sure the copied binaries were executable, shown in Step 6.

Configuring Grafana Loki

A sample configuration file for Loki is supplied with the source code, in the file /cmd/loki/loki-local-config.yaml. In our example we are using this supplied sample configuration file, which has Loki writing its indexes to a temporary location and listening on the standard port. In your installation you may wish to configure Loki to use a different data disposition location, use a different port, or configure TLS. Follow the Loki documentation to alter the example configuration to suit your needs.

We copied the sample config file to /usr/local/etc for convenience, and created a systemd unit file to run our Loki instance. The systemd unit was created at /usr/local/lib/systemd/system/loki.service with the content shown at Example 7-70.

Example 7-70 Systemd unit file for Loki

```
[Unit]
Requires=network-online.target
After=network-online.target named-chroot.service
[Service]
Type=exec
ExecStart=/usr/local/bin/loki -config.file=/usr/local/etc/loki-local-config.yaml
-log.level=warn
RemainAfterExit=false
StandardOutput=journal
[Install]
WantedBy=default.target
```

We started Loki from the command line with the following commands:

```
$ sudo systemctl daemon-reload
$ sudo systemctl enable --now loki.service
Created symlink /etc/systemd/system/default.target.wants/loki.service ?
/usr/local/lib/systemd/system/loki.service.
$ sudo systemctl status loki
```



```

• loki.service
  Loaded: loaded (/usr/local/lib/systemd/system/loki.service; enabled; preset: disabled)
  Active: active (running) since Mon 2024-03-25 02:53:57 EDT; 7s ago
  Main PID: 187082 (loki)
  Tasks: 10 (limit: 100835)
  Memory: 88.9M
  CPU: 117ms
  CGroup: /system.slice/loki.service
          ??187082 /usr/local/bin/loki
-config.file=/usr/local/etc/loki-local-config.yaml -log.level=warn

Mar 25 02:53:57 rdbkpmr1.pbm.ihost.com systemd[1]: Starting loki.service...
Mar 25 02:53:57 rdbkpmr1.pbm.ihost.com systemd[1]: Started loki.service.

```

For testing we could have simply launched the Loki process from the command line and then sent it to the background, but we demonstrate here a more production-like scenario with a systemd service.

Configuring Promtail to process log data

A number of sample Promtail configuration files are supplied in the source code, in the location `./clients/cmd/promtail/`. In this location, the file `promtail-local-config.yaml` provides a simple example of reading local log files in the `/var/log/` directory. There is also a sample file that configures Promtail to read from `systemd-journal`.

Note: Promtail documentation states that systemd journal integration only functions on AMD64 systems. However, we found that it did work properly on s390x as well.

Since the choice of log files to be processed may be different between locations, it is important to configure Promtail according to your installation's needs.

Promtail scraping and pipelines

The Promtail configuration file specifies two important aspects of Promtail operation:

- ▶ Scrapers, that indicate where log data is to be sourced from, and
- ▶ Pipelines, that control any required formatting or transformation of messages.

Pipelines can be used to parse the message into fields that can either be used directly as a labels, or manipulated in some way (by another pipeline stage) before being used either as a label or as the message output to be sent to Loki.

A simple example of a pipeline would be to take messages in Apache log format and parse them into timestamp, requester IP/hostname, and request type and content fields. The request type could then be used as a label, which Loki would index. This would then allow messages to be queried by the request type in Grafana.

Note: It might be tempting to add the request origination IP address (or hostname) as a label. The Grafana Loki documentation recommends against this, however, because a very busy website would have a very large number of requester IP addresses in its log. Labels that have a very large number of values cause the index storage in Loki to increase dramatically, increasing Loki resource requirements and negatively impacting performance.

Not all fields have to be indexed to allow searching. The free-text filter capability (which we will see later) is enough to allow particular log lines, such as requests from given IP addresses or hostnames in this example, to be searched-for in the logs without the need to label the field.

It might also be useful in this case to parse out the HTTP status code and transform the output by adding the plain-text explanation of the status code alongside the code in the message output.

Note: Any manipulation of the message text would only be visible in the Loki content via the Grafana interface. Promtail does not modify the source log files in any way.

The full explanation of Promtail configuration is outside the scope of this book. For more information about writing pipelines and configuring Promtail, refer to <https://grafana.com/docs/loki/latest/send-data/promtail/configuration/>.

Note: Grafana documentation recommends that syslog hosts do **not** send their messages directly to the network port of the Promtail syslog scraper. There are at least two reasons for this:

- ▶ Promtail understands the syslog format documented in RFC 5424, but there are many hosts that implement older syslog daemons that have not been updated. For consistency of message handling, it is recommended to retain a traditional syslog daemon with better capability to handle messages in older formats.
- ▶ Large installations may have thousands of systems sending syslog messages, including systems still using UDP (instead of TCP). Traditional syslog daemons are designed to support such environments, by minimizing (or eliminating) processing of messages, and by using threading, buffering, and other techniques to ensure messages are not lost. Promtail, on the other hand, does some amount of processing of **every** message to set labels (and may do more if message transformation is set up). This additional processing may result in delay or loss of messages. A syslog daemon as the first receiver can operate as a “store-and-forward” capability to ensure that all messages are captured and destaged to Promtail as resources allow.

Our example Promtail configuration

We created an example Promtail configuration file which is shown in Example 7-71. For our example, we created a Promtail configuration from scratch to use the syslog scraper, which natively supports receiving messages using the syslog protocol. We then configured our central logging server to forward messages to Promtail. We also make sure that important syslog fields like facility and level, as well as hostname, are marked as labels so that we can search for logs in a syslog-relevant way using Grafana.

Example 7-71 Example Promtail configuration file

```
server:  
  http_listen_port: 9081
```

```

    grpc_listen_port: 0

positions:
  filename: /tmp/positions-syslog.yaml 1

clients:
  - url: http://localhost:3100/loki/api/v1/push 2

scrape_configs: 3
  - job_name: syslog
    syslog: 4
      listen_address: 127.0.0.1:20514 5
      label_structured_data: false
      labels: 6
        job: syslog 7
        use_incoming_timestamp: false 8
        idle_timeout: 12h 9
    relabel_configs: 10
      - source_labels: [__syslog_message_hostname]
        target_label: hostname
      - source_labels: [__syslog_message_severity]
        target_label: level
      - source_labels: [__syslog_message_app_name]
        target_label: application
      - source_labels: [__syslog_message_facility]
        target_label: facility
      - source_labels: [__syslog_connection_hostname]
        target_label: connection_hostname

```

Descriptions of these configuration items:

1. Promtail keeps a data file that records where it is up to in reading log file data. This is likely unnecessary for the syslog scraper since Syslog data arrives in a stream, but this section seems to be a standard part of Promtail configuration so we retained it. The path given must be different from other Promtail instances to avoid any conflicts between multiple Promtail processes.
2. In Promtail configuration, clients are the Loki instances that Promtail sends log data to. The URL of the API endpoint of our Loki instance is given here.
3. The `scrape_configs` section defines the scraper operated by this Promtail instance.
4. This line indicates that a syslog scraper is being defined.
5. The scraper will listen for Syslog messages at the address and port given here.
6. The labels section defines any hard-coded labels that will be passed to Loki by this scraper. It is Promtail convention to pass a label called “job” that refers to the identifier of the Promtail instance. This makes it easy to search among messages obtained from a given Promtail scraper.
7. In some installations it will be appropriate to preserve the timestamps that are received in the syslog message. This will be the case if all of your systems are synchronised to a consistent NTP time source, for example. You would do that by setting this value to `true`. In our lab, our syslog hosts do not have uniform time. We found that messages from some remote hosts were appearing out of order. We set this option to `false` to use a timestamp generated by Promtail or Loki when messages arrive. That way, when we viewed messages from different hosts together in a single Grafana search, the messages appeared in correct chronological order.

8. In some of the screen shots you will see later, you might notice some messages logged by rsyslogd about being disconnected from Promtail. We found that the default setting of `idle_timeout` is too short and causes Promtail to drop the syslog connection. Setting this value to a number that is in excess of the usual interval of syslog messages is required to make sure that the connection stays up and does not generate unneeded log noise.
9. The Syslog scraper understands the format of syslog messages and generates a number of labels containing the metadata of messages. The `relabel_configs` section specifies how the system-generated labels are exposed and mapped to human-friendly names.
10. In our environment we considered removing this label, because all of the syslog messages received by Promtail come from the same syslog server (Rsyslog on our central log server) so this label only has one value. If you have multiple syslog servers sending messages to Promtail, however, you could retain this mapping to allow for querying messages from different source syslog servers.

We created this configuration file as `/usr/local/etc/promtail-itso-config.yaml`. We also copied the sample `systemd-journal` config file from the distribution source to `/usr/local/etc`.

We then configured two instances of Promtail on our log server. The first was to gather logs from the local `systemd-journal` process (using the copied example configuration file as-is), and the second was to receive Syslog data from our central Rsyslogd log server (using our customized configuration file). Since the only thing that differs between the two Promtail instances is the configuration file, we used one `systemd` template unit which gets instantiated for each Promtail instance. The `systemd` unit file is shown in Example 7-72.

Example 7-72 Systemd unit file for Promtail instances

```
[Unit]
Requires=network-online.target loki.service
After=network-online.target named-chroot.service loki.service
[Service]
Type=exec
ExecStart=/usr/local/bin/promtail -config.file=/usr/local/bin/promtail-%i.yaml
-log.level=error
RemainAfterExit=false
StandardOutput=journal
[Install]
WantedBy=default.target
```

The “%i” in the unit file is replaced by whatever text appears after the “@” character in the name provided for the desired unit. Our example creates a configuration that automatically expands the correct configuration file based on the unit name requested. This way, we can create new Promtail instances simply by creating a new configuration file and calling for a `systemd` unit with the so-named Promtail configuration file.

We started Promtail from the command line, and checked the status, with the commands shown in Example 7-73.

Example 7-73 Starting Promtail

```
$ sudo systemctl daemon-reload
$ sudo systemctl enable --now promtail-@journal.service
Created symlink /etc/systemd/system/default.target.wants/promtail-@journal.service ?
/usr/local/lib/systemd/system/promtail-@.service.
$ sudo systemctl enable --now promtail-@itso-config.service
Created symlink /etc/systemd/system/default.target.wants/promtail-@itso-config.service ?
/usr/local/lib/systemd/system/promtail-@.service.
$ sudo systemctl status promtail-@journal.service
```

```

• promtail-@journal.service
  Loaded: loaded (/usr/local/lib/systemd/system/promtail-@.service; enabled; preset:
disabled)
  Active: active (running) since Mon 2024-03-25 03:35:04 EDT; 10min ago
  Main PID: 188729 (promtail)
  Tasks: 10 (limit: 100835)
  Memory: 23.0M
  CPU: 828ms
  CGroup: /system.slice/system-promtail\x2d.slice/promtail-@journal.service
          ??188729 /usr/local/bin/promtail
  -config.file=/usr/local/etc/promtail-journal.yaml -log.level=error

Mar 25 03:35:04 rdbkpmr1.pbm.ihost.com systemd[1]: Starting promtail-@journal.service...
Mar 25 03:35:04 rdbkpmr1.pbm.ihost.com systemd[1]: Started promtail-@journal.service.
$ sudo systemctl status promtail-@itso-config.service
• promtail-@itso-config.service
  Loaded: loaded (/usr/local/lib/systemd/system/promtail-@.service; enabled; preset:
disabled)
  Active: active (running) since Mon 2024-03-25 03:40:15 EDT; 5min ago
  Main PID: 188918 (promtail)
  Tasks: 9 (limit: 100835)
  Memory: 19.0M
  CPU: 269ms
  CGroup: /system.slice/system-promtail\x2d.slice/promtail-@itso-config.service
          ??188918 /usr/local/bin/promtail
  -config.file=/usr/local/etc/promtail-itso-config.yaml -log.level=error

Mar 25 03:40:15 rdbkpmr1.pbm.ihost.com systemd[1]: Starting
promtail-@itso-config.service...
Mar 25 03:40:15 rdbkpmr1.pbm.ihost.com systemd[1]: Started promtail-@itso-config.service.

```

After we started the Promtail process for Syslog, we configured our central log server to forward all captured log messages to our Promtail syslog scraper. We added the configuration lines shown in Example 7-74 to our `/etc/rsyslogd.d/logserver.conf` file.

Example 7-74 New configuration lines

```

(ruleset="promtail"){
  *.* action(type="omfwd" protocol="tcp" target="127.0.0.1" port="20514"
Template="RSYSLOG_SyslogProtocol23Format" TCP_Framing="octet-counted" KeepAlive="on")
}
call promtail

```

The line “`call promtail`” in Example 7-74 block adds the definition of the Promtail rule set to the default ruleset in Rsyslogd. We also added the same line inside the definition of the ruleset that handles the messages received from remote syslog hosts, so that those messages also get forwarded to Promtail.

After restarting the Rsyslog process, we checked to see that Rsyslog was connected to Promtail (Example 7-75).

Example 7-75 Verify Rsyslog connection

```

# systemctl restart rsyslog.service
# ss -apnt | grep 20514
LISTEN 0    4096      127.0.0.1:20514      0.0.0.0:*      users:(("promtail",pid=299896,fd=8))
ESTAB  0      0        127.0.0.1:36986     127.0.0.1:20514 users:(("rsyslogd",pid=299920,fd=8))
ESTAB  0      0        127.0.0.1:20514     127.0.0.1:36986 users:(("promtail",pid=299896,fd=11))
#

```

Note: Port 20514 was not chosen arbitrarily. We first tried a different port (5514), but found that rsyslogd was prevented by SELinux from connecting to Promtail.

As described in “Server configuration” on page 183, only certain ports are part of the supplied RHEL SELinux profile for use by syslog daemons, and 20514 is one of the permitted ports. We chose to switch our Promtail process to 20514 to avoid changing SELinux. In your installation you could also decide to define the port you choose for Promtail to SELinux (like we did in the Rsyslog configuration above).

Viewing log data using Grafana

To view the log data using Grafana, you need a running Grafana instance in your installation. If you have an existing Grafana deployment, you could simply add your Loki instance as a data source to Grafana. Skip to “Adding Loki as a data source to Grafana” on page 200 to see this.

If you do not have an existing Grafana instance, you can install Grafana on a host in your Linux on IBM Z environment. You could install Grafana on your central log server, for example. We chose to do this in our lab environment.

Grafana is included in RHEL, so you can use **dnf** to install it. Once installed, make sure the firewall is configured to accept connections and start the service. This is shown in Example 7-76.

Example 7-76 Installing and activating Grafana on RHEL 9

```
[root@rdbkpmr1 ~]# dnf install grafana
Updating Subscription Management repositories.
Last metadata expiration check: 2:08:27 ago on Wed 20 Mar 2024 12:31:14 AM EDT.
Dependencies resolved.
=====
Package      Architecture Version      Repository                               Size
=====
Installing:
grafana       s390x        9.2.10-7.el9_3 rhel-9-for-s390x-appstream-rpms      72 M
Installing weak dependencies:
grafana-pcp   s390x        5.1.1-1.el9     rhel-9-for-s390x-appstream-rpms     9.9 M

Transaction Summary
=====
Install 2 Packages

Total download size: 82 M
Installed size: 346 M
Is this ok [y/N]: y
Downloading Packages:
(1/2): grafana-pcp-5.1.1-1.el9.s390x.rpm      4.2 MB/s | 9.9 MB  00:02
(2/2): grafana-9.2.10-7.el9_3.s390x.rpm      6.7 MB/s | 72 MB  00:10
-----
Total                                          7.6 MB/s | 82 MB  00:10
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
```

```

Running scriptlet: grafana-9.2.10-7.el9_3.s390x      1/2
Installing       : grafana-9.2.10-7.el9_3.s390x      1/2
Running scriptlet: grafana-9.2.10-7.el9_3.s390x      1/2
Installing       : grafana-pcp-5.1.1-1.el9.s390x      2/2
Running scriptlet: grafana-pcp-5.1.1-1.el9.s390x      2/2
Verifying        : grafana-pcp-5.1.1-1.el9.s390x      1/2
Verifying        : grafana-9.2.10-7.el9_3.s390x      2/2
Installed products updated.

```

```

Installed:
  grafana-9.2.10-7.el9_3.s390x          grafana-pcp-5.1.1-1.el9.s390x

```

```

Complete!
[root@rdbkpmr1 ~]# systemctl enable --now grafana-server.service
Created symlink /etc/systemd/system/multi-user.target.wants/grafana-server.service.
? /usr/lib/systemd/system/grafana-server.service.
[root@rdbkpmr1 ~]# firewall-cmd --add-service=grafana --permanent
success
[root@rdbkpmr1 ~]# firewall-cmd --reload
success
[root@rdbkpmr1 ~]#

```

Once this is done, you can use your browser to access the Grafana interface at `http://<hostname>:3000`, which should look like Figure 7-10.

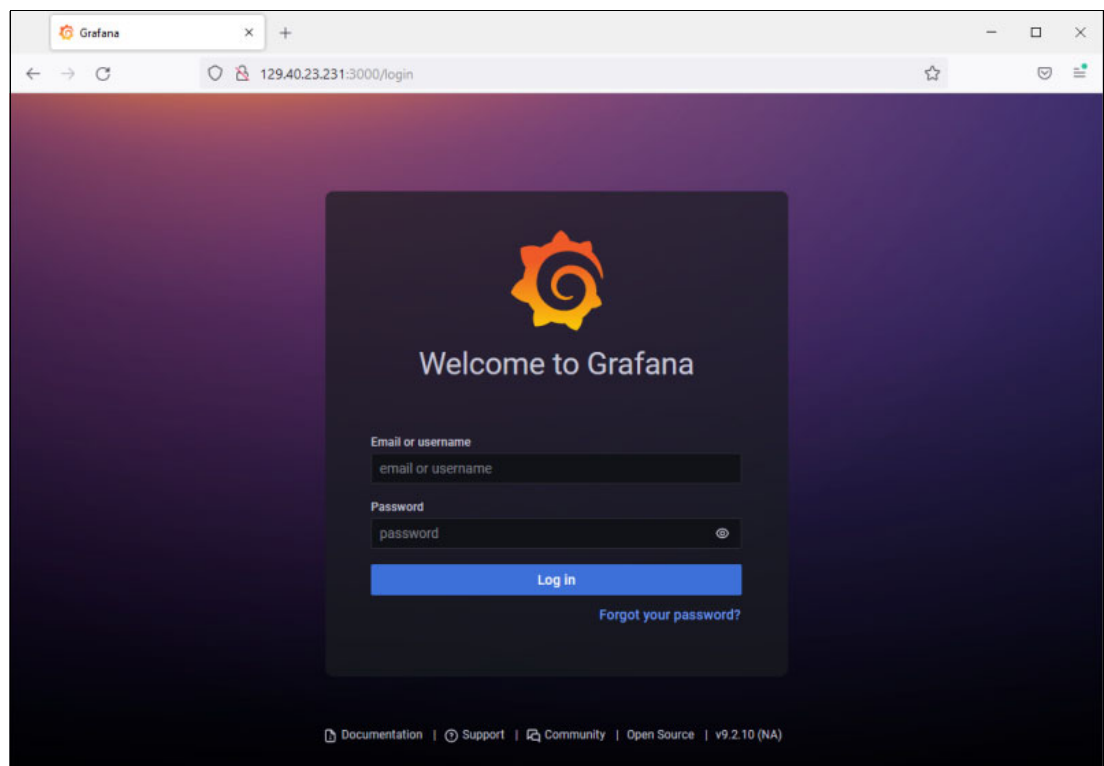


Figure 7-10 Grafana login panel

The default ID and password for a new Grafana Installation is **admin/admin**, and when you log on with these credentials you will be prompted to reset the password.

Note: We recommend that you configure your Grafana instance to use centralized authentication such as LDAP or Active Directory. This configuration is out of scope for this book, but it is important that you do this as one of your first setup tasks for your new Grafana instance!

Once you have changed the admin password, you will see the main Grafana interface. From here you can proceed to add your Loki data source to Grafana and start viewing log data.

Adding Loki as a data source to Grafana

To view log data in our Grafana instance, we added Loki as a data source to Grafana.

Note: Loki runs on TCP port 3100 by default. If you are running your Grafana interface on a different system than your central log server (or where you are running Loki) you will need to make sure Grafana can connect to Loki on port 3100. This means any firewalls between them must allow TCP port 3100 from Grafana to Loki.

In the Grafana interface, we clicked the gear icon for “Configuration” in the lower-left of the screen (you can also hover on the gear and select “Data sources” in the pop-up menu). The configuration panel appeared, with the Data sources tab selected, as shown in Figure 7-11.

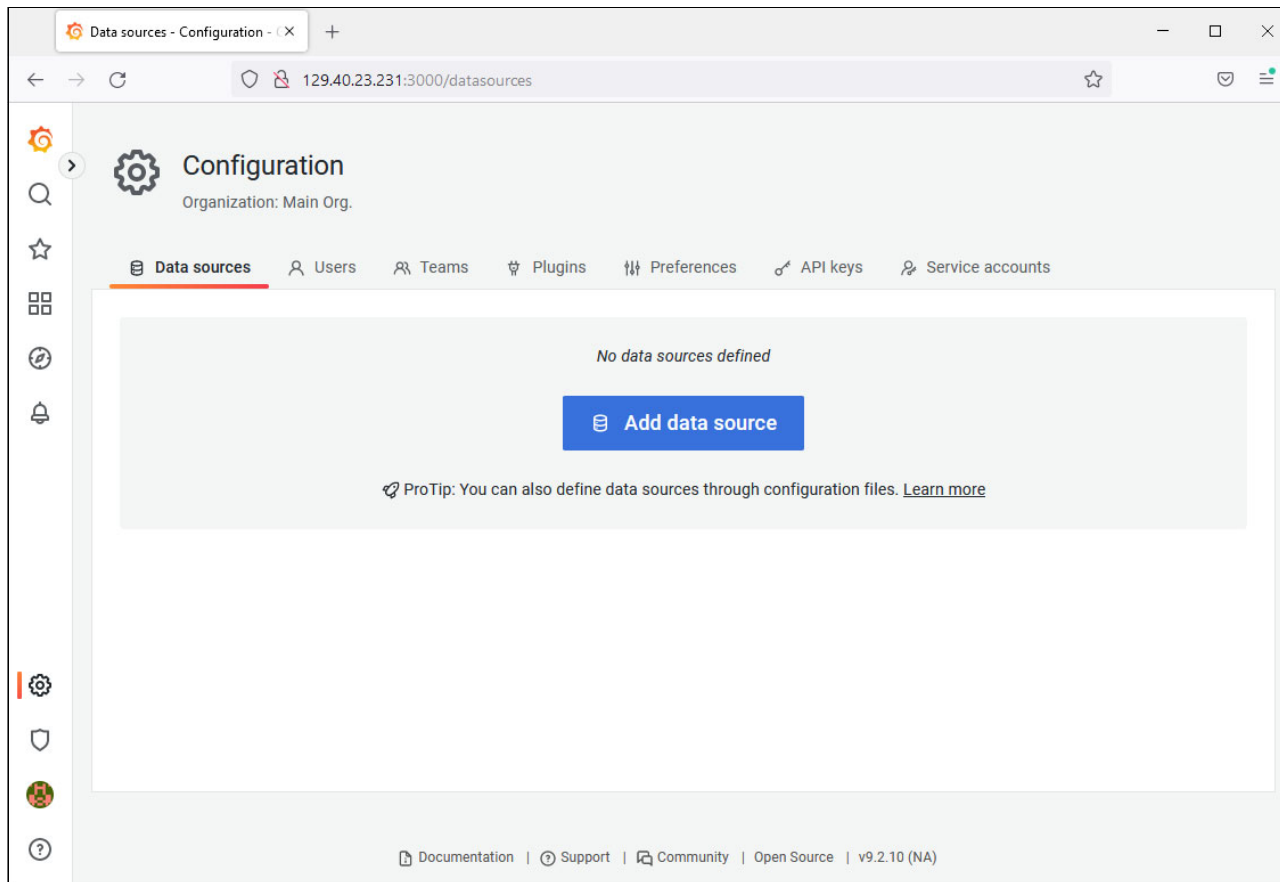


Figure 7-11 Grafana Configuration panel

We clicked on **Add data source** to display the options for adding a data source. Scrolling down the list until we saw “Loki”, we clicked on it to display the options for adding a Loki data source as shown in Figure 7-12 on page 201.

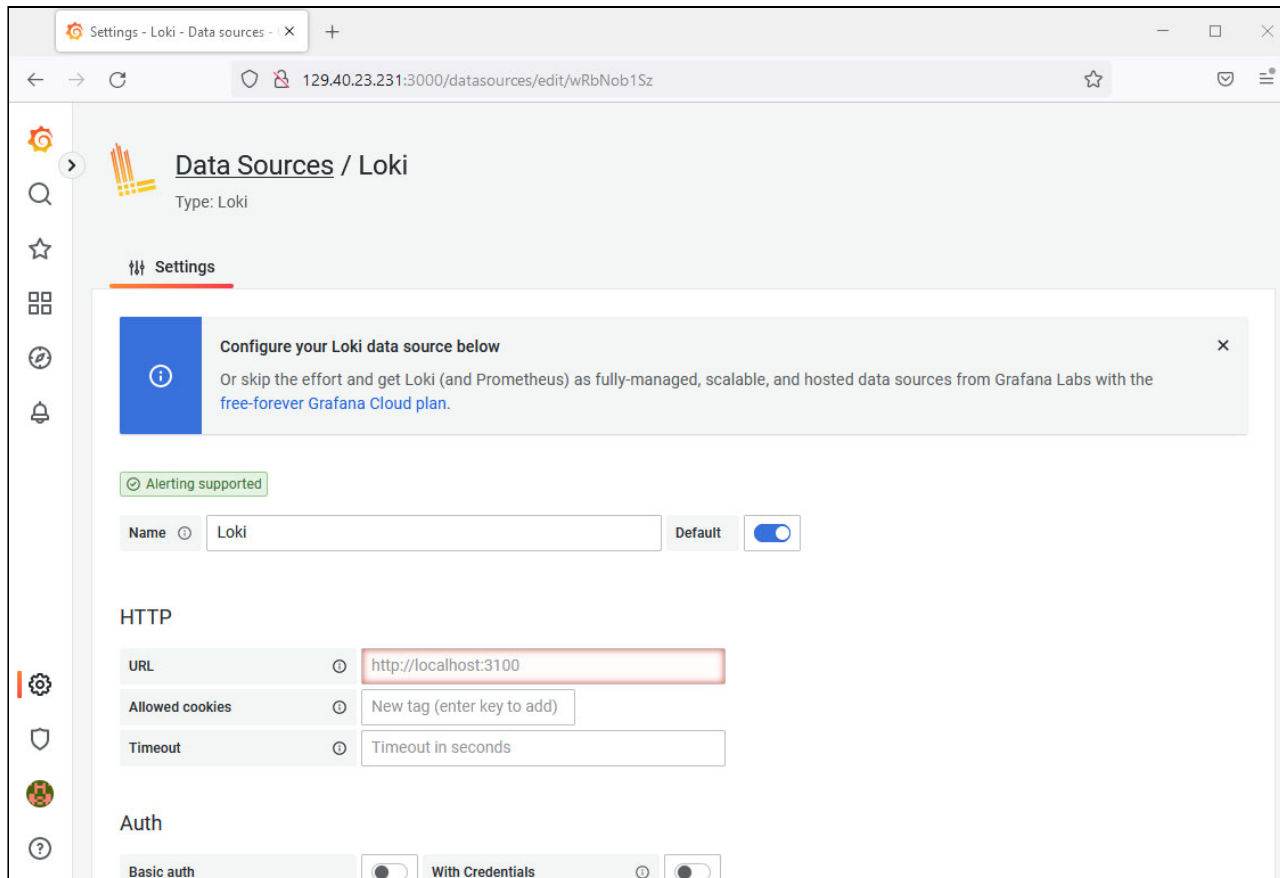


Figure 7-12 Grafana Add Data Source Loki

The only value we needed to add to this panel was the URL for our Loki data source. As per our configuration, the default of `http://localhost:3100` was suitable, but we had to type it into the field because the text we saw on the screen was just a visual prompt.

Note: As you can tell from the “HTTP” URL, our configuration did not use TLS for connection security, nor did we configure any credentials for authentication. This is an example configuration for testing only! The only security present is that the Promtail and Loki processes only listen on the localhost IP address (127.0.0.1 or ::1) and are not directly accessible from outside the server. Realistically though, an administrator with SSH access to the server could use SSH tunneling to bypass this.

We strongly recommend that any serious or production implementation of this solution use at least an authentication method. In addition, if the Promtail and/or Loki processes will be accessible from other hosts, TLS must be used to protect data in transmission.

Once we filled in the URL for our Loki instance, we scrolled to the bottom of the page and clicked the “Save and test” button to commit the definition of our Loki data source. After a very brief instant, Grafana confirmed that the changes had been saved and also confirmed that the data source was valid with the confirmation message shown in Figure 7-13.



Figure 7-13 Grafana Add Data Source Loki -- confirmation

We received the “labels found” message because Promtail was already configured and forwarding messages to Loki. If we had not already configured Promtail, there would have been no content in Loki and we would have receive a warning instead. The warning would have informed us that the data source was connected, but since no labels were found we should check that the data source is correctly set up.

Note: If you receive that warning when you define the Loki data source to Promtail, there may not be a problem. Loki discovers and builds indexes based on the data it receives from Promtail, so if you added your Loki data source to Grafana before doing the Promtail part you can expect that Loki would have no labels and you would get the warning.

We recommend doing the configuration steps in the same order we did, so that the “labels found” message gives you some confirmation that your Promtail and Loki instances are working together correctly.

Once Loki is defined to Grafana, we used the **Explore** tab in Grafana to view our log data. The initial screen is shown in Figure 7-14. Grafana builds the fields on this screen (along with many others) dynamically based on the content it queries from the data sources. This can be another good way to verify the configuration (especially if the configuration involves any pipeline-based transformation of messages in Promtail).

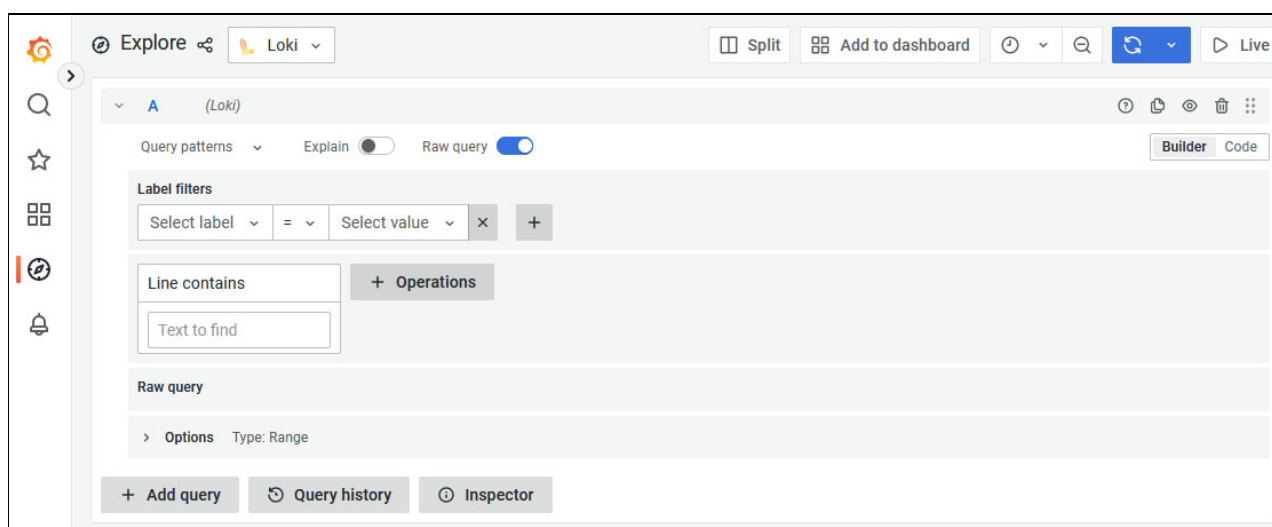


Figure 7-14 Grafana Explore function

We clicked on the **Select label** item, and Grafana filled in the list of available labels based on what it queried from Loki. In our installation, we saw labels obtained from both the syslog and systemd Promtail instances, as shown in Figure 7-15 on page 203.

The **job** label corresponds to the instance of Promtail that generated the message in Loki. Choosing the **job** label allows us to query against a subset of messages based on the value of the **job** label. In our installation, to see all of the messages that have been forwarded from Syslog, for any host, we firstly selected the **job** label. Grafana then queries Loki for values of the **job** label, and uses that information to fill in the **Select value** item.

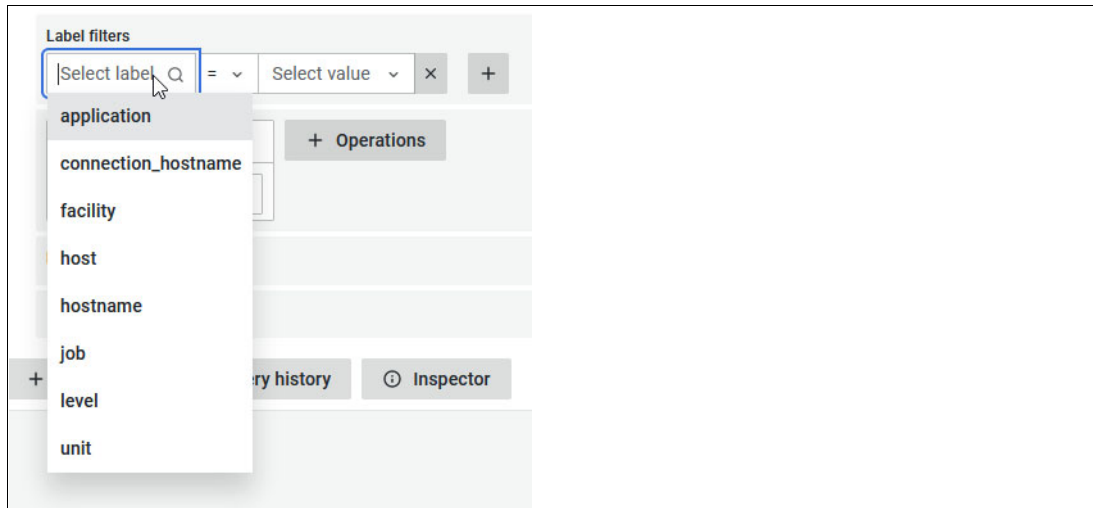


Figure 7-15 Available labels from Loki

The resulting available values on our system can be seen in Figure 7-16. We then select the value **syslog** to view messages that came from syslog.

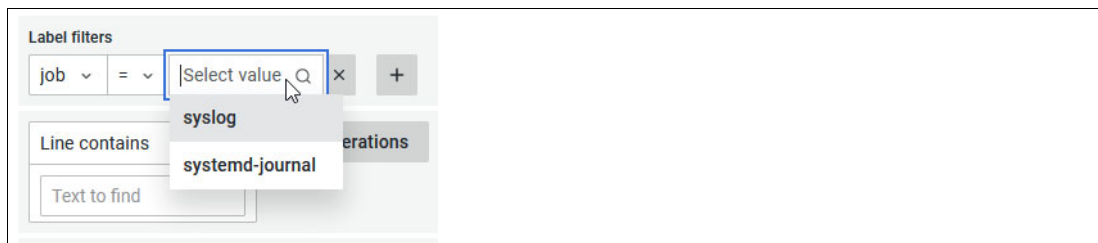


Figure 7-16 Available values after selecting “job” label

Once we selected **syslog**, the **Raw query** field showed `{job="syslog"} | = ```. This is the LogQL query that would be executed by Grafana against Loki if we executed the query. LogQL is a powerful query language used by Grafana and other similar tools to make it easy to query log sources like Loki. Operators who are experienced with LogQL can generate sophisticated queries very quickly.

To run the generated query, we pressed **Run query**. Log data from syslog was presented, along with a graph showing the timing of messages and the message type. We scrolled down the screen to produce the screenshot shown in Figure 7-17 on page 204.

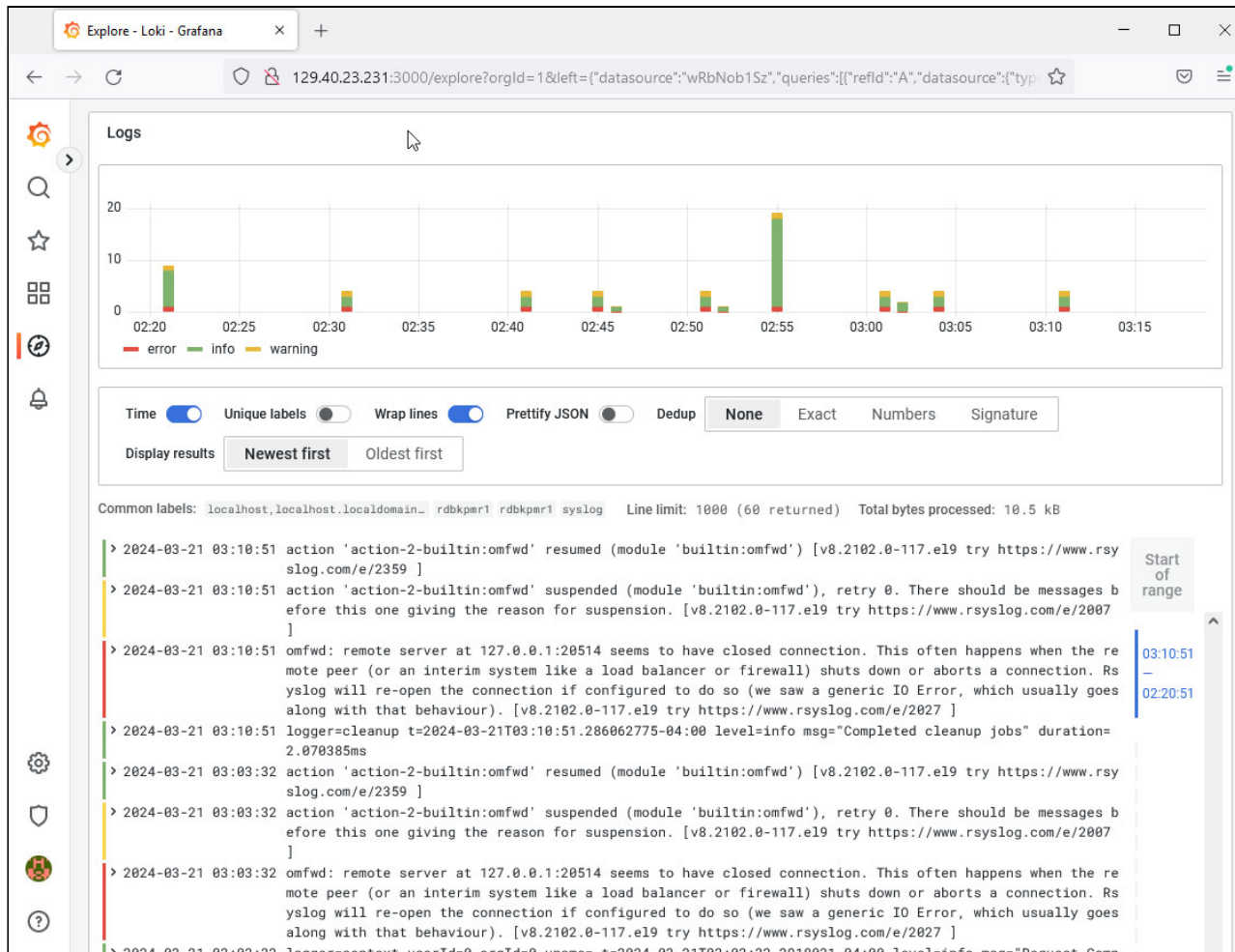


Figure 7-17 Syslog query result

By changing the query options, it is possible to filter the display of messages. For example, to see only the messages with the word “action” in the text we clicked on “Operations”, and selected **Line filters / Line contains** to display a filter field. We then entered “action” in the **Line contains** field and re-ran the query. The result of this is shown in Figure 7-18 on page 205.

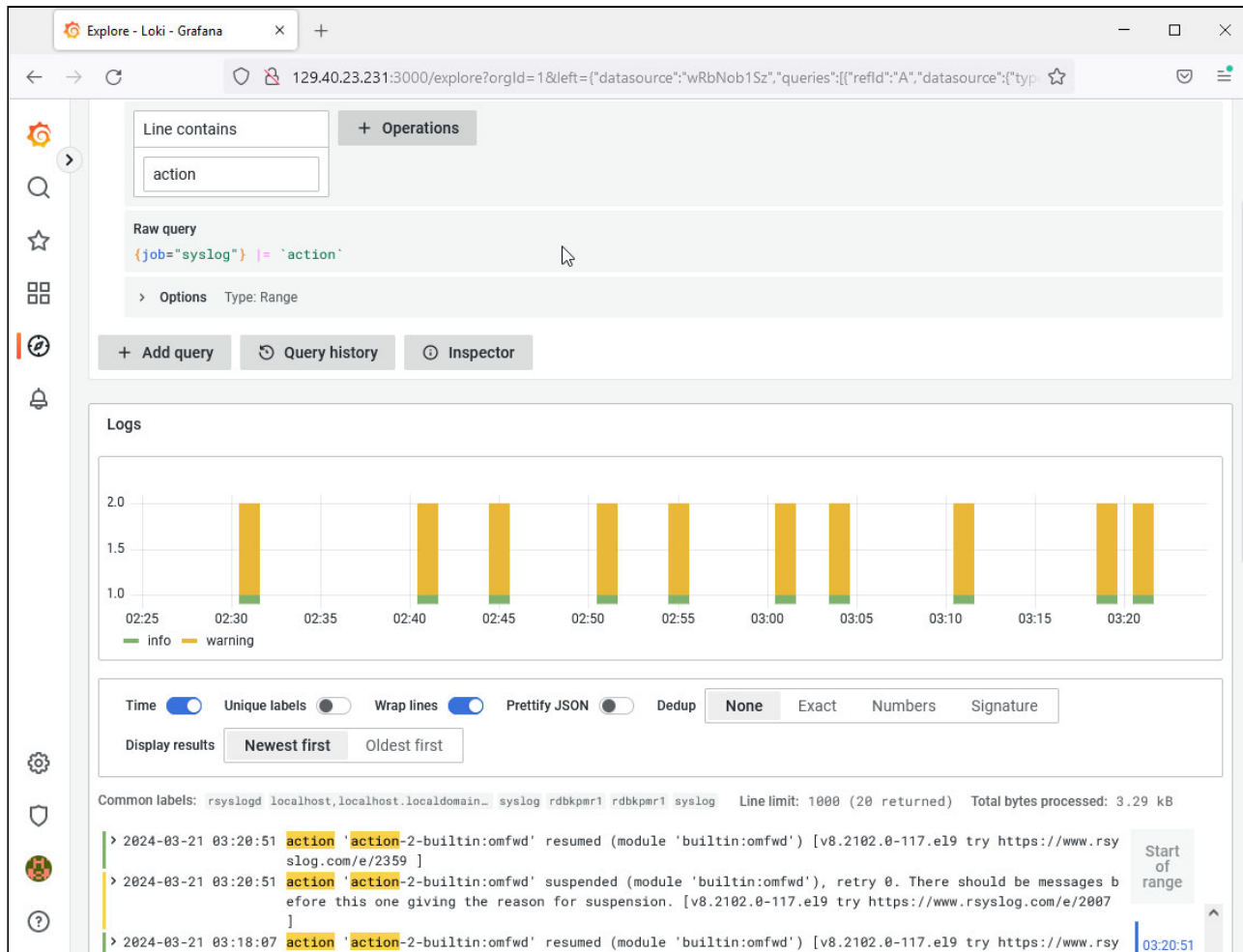


Figure 7-18 Searching for text in log messages

Pressing **Live** (the gray button with a triangle, to the immediate right of the blue Query button) put the log display into a live display mode. In this mode, new log messages appeared as soon as Loki received them from Promtail. We could pause or stop the live display using the buttons that replaced the Live button on the display.

Once a useful query has been created, it can be added to a dashboard. Dashboards are one of the most useful features of Grafana, allowing customized views of data from different sources to be displayed in a single pane. Organizations using Grafana to display Prometheus or InfluxDB metrics data would be familiar with building concise displays of that metrics data, and with Loki they would be able to add a relevant log stream to the same display.

Data retention

When Grafana Loki is implemented in an organization, data retention will be a key consideration. Our example stores logs in the Loki instance on a temporary basis only, and does not replace the retention of the original log files on our central log server. If Loki is being used purely as a visual access method for recent log files only, this might be acceptable. On the other hand, the extensive searching capability of the Grafana interface (and LogQL) means that an organization may prefer to retain the log data in Loki instead, and use the central log server as a “store-and-forward” only (or still keep the central log server’s log files on disk, but retain them for a much shorter time).

It is up to the organization deploying Loki to decide what policy or policies to adopt. Loki supports a variety of storage types, from filesystem through to cloud storage (such as Amazon S3 and IBM Cloud Object Storage), and can be configured to manage storage needs through deduplication of log data as well as controlling retention and deletion.

Grafana Loki is a powerful log aggregation tool with many more capabilities than we can describe in the scope of this book.

7.8 Deploying Samba

Samba is an open software suite that runs the Server Message Block (SMB) protocol over the Internet Protocol network and provides seamless file and print services to users. Although there are several similar commercial products available, Samba is the implementation that is most commonly used in Linux environments to share files and printers. It is available for Linux on IBM Z from both Red Hat and SUSE and allows interoperability between UNIX/Linux servers and Windows/Linux based clients. Samba runs easily on Linux on IBM Z because IBM Z has fast I/O that provides high performance access to applications and files.

Before deploying Samba, ensure that appropriate analysis and planning has been performed before any migration activity. The checklists provided in this book have been created to help identify the many considerations that should be made which will help prevent problems during migration.

In our sample scenario, the z/VM guest has already been set up and a minimal Linux operating system has been installed. The Linux guest is named LNSUDB2, and has SUSE Enterprise Linux Server11 SP3 installed with one virtual CPU and 1 GB of virtual memory. Like LDAP, a Samba server typically does not require a large amount of CPU or RAM to run on Linux on IBM Z. It is presumed that an adequate RPM repository installation source is already set up and available for the installation of the application software that will be used.

For more information, see [Samba](#).

This example will be a stand-alone server with a local, non-replicated directory service. Nevertheless, migrating an existing Samba installation on x86 to Linux on IBM Z should be straight forward.

7.8.1 Installing Samba software

Installing the software is relatively easy.

- To install Samba on SUSE Linux Enterprise Server:

Run **zypper install samba** to install the Samba and its dependencies packages on SUSE Enterprise Linux Server.

- To install Samba on Red Hat Enterprise Linux:

Run **yum install samba** to install the Samba and its dependencies packages on RHEL.

7.8.2 Configuring SAMBA

In this section, we describe how to configure Samba first on SUSE Enterprise Linux Server and then on RHEL.

Configuring file server on SAMBA on using YaST

All of the activities to create a working configuration are facilitated by the Samba server YaST module. By following a few simple panels in YaST, the SAMBA services can be configured and running in short order.

To configure a Samba server, start YaST and select **Network Services** → **Samba Server** and complete the fields that are shown in Example 7-77 with your network information.

Example 7-77 Initial configuration of Samba on YaST

Workgroup name or Domain Name: Select your existing name from the Workgroup or Domain
Samba Server Type (PDC, BDC or Stand Alone): Specify whether your server should act
Start service : To start after the server reboot, choose *during the boot*
Firewall Settings : If you are running the firewall servers inside this server, mark the option Open Port in Firewall
Samba root Password: choose a password for your Samba service

After the initial installation step is completed, confirm by selecting **OK**. You will be able to change the settings later in the Samba configuration dialog on YaST, as shown in Figure 7-19.

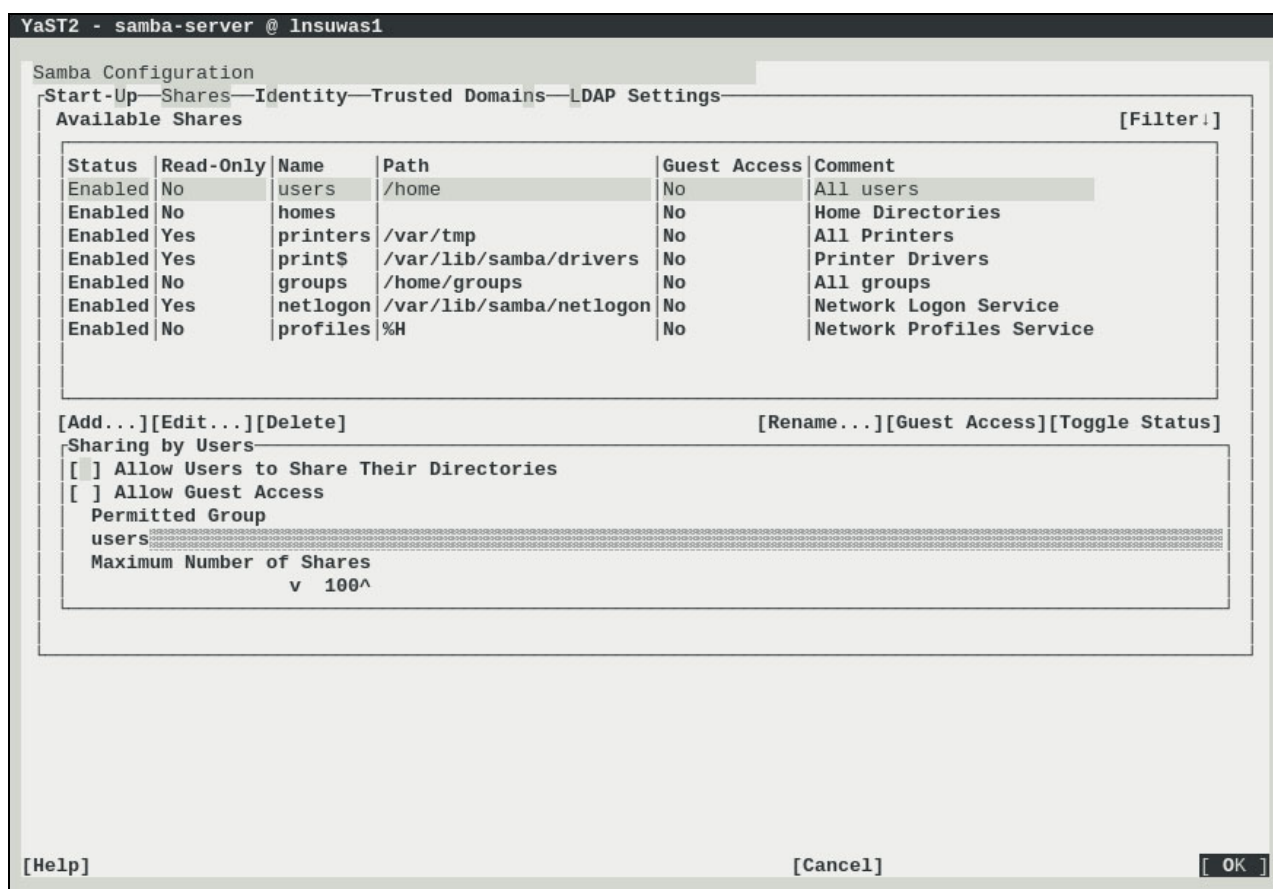


Figure 7-19 Samba Server Configuration tab

Samba shares

In order to share resources such as folders or printers on a Samba server, you must first identify these “shares”. You can configure your shares on YaST in the Samba Server.

In the Samba configuration tab, shown in Figure 7-19 on page 207, select the **Shares** tab and then select **Add**. Provide the information shown in Example 7-78 and shown on the YaST2 panel in Figure 7-20.

Example 7-78 Creating new share on Samba using Yast

Share Name : Fill out the share name
Share Description : brief description of the share
Share Type : select if you are sharing a folder or a printer
Share Path : browser the folder name. Make sure that the folder is set up with the correct permission on the Linux filesystem
Select If you need Read only access and Inherit the config to the subdirectories

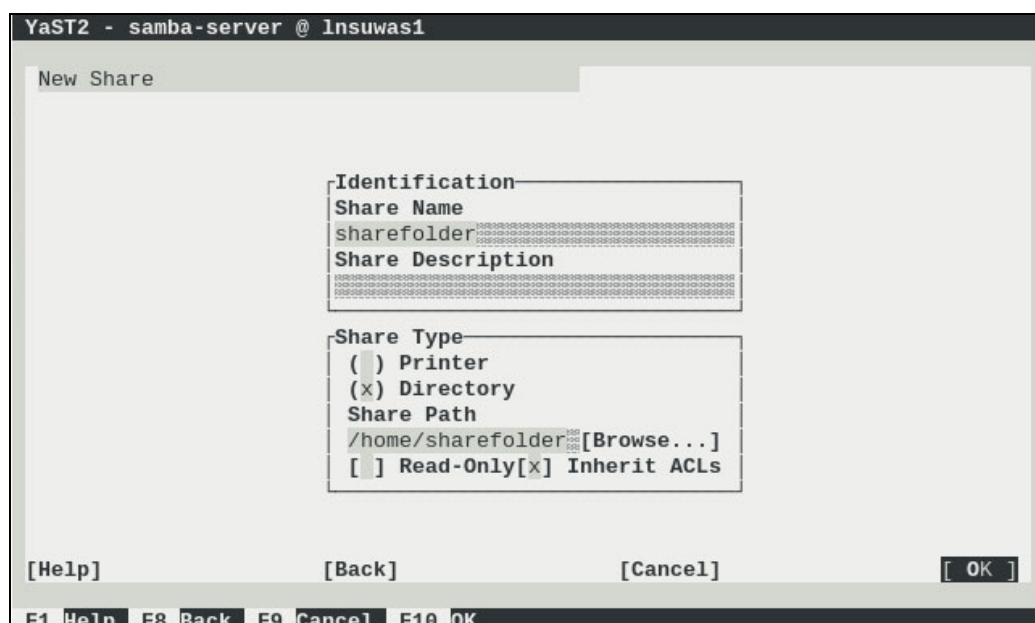


Figure 7-20 Sample of creating new share on Samba using Yast

After completing this task, you should be able to map to the server from your client machine.

Note: If you intend to use basic Linux authentication, that is, using the *passwd* file, you must change the Samba user password using the command `smbpasswd -a <userid>`.

LDAP settings for Samba

Many companies use LDAP to provide a single signon, where a password for one user is shared between services. The activities to create a working configuration are facilitated by the OpenLDAP server YaST module. Although the LDAP configuration on SAMBA is an important feature, the configuration is out of the scope of this book.

For more information on how to set up LDAP on Samba, visit [Samba](#).

Configuration files

You can manually set up configuration files for Samba. The main configuration file on SUSE Enterprise Linux Server is stored in `/etc/samba/smb.conf`. You will find two sections:

- ▶ [global] to general settings
- ▶ [share] to specify specific settings about sharing files and printers

For more information about Samba configuration on SUSE Enterprise Linux Server, see the SUSE Linux Enterprise Server 11 Administration Guide, [Samba](#) section.

Note: RHEL uses the same structure as SUSE Enterprise Linux Server 11 for the Samba main configuration files.

Starting and stopping the Samba service

The *smb* service controls the server daemon and can be stopped and started by the commands as shown in Example 7-79.

Example 7-79 Stopping and starting the Samba service

```
To stop and start the service both SUSE Enterprise Linux Server and RHEL:  
Stop the service: /etc/init.d/smb stop  
Start the service: /etc/init.d/smb start
```

7.9 Deploying Terraform

HashiCorp [Terraform](#) is a popular open-source Infrastructure as Code (IaC) tool. It allows users to define and provision infrastructure resources using a declarative configuration language. Terraform then translates these configuration files into an execution plan, which outlines the actions it will take to achieve the desired state. This plan can be reviewed before applying changes to the infrastructure. Once approved, Terraform applies the plan and provisions or updates the infrastructure accordingly. It manages the entire life cycle of infrastructure resources, from creation to modification to deletion.

The following steps will guide you in deploying Terraform by using a Dockerfile.

1. Ensure that Docker is running and ready to be used.
2. Download the Terraform Dockerfile, as shown in Example 7-80.

Example 7-80 Download Terraform Dockerfile

```
[root@y1sprd prd]# sudo mkdir /root/Terraform  
[root@y1sprd prd]# wget  
https://raw.githubusercontent.com/linux-on-ibm-z/dockerfile-examples/master/Terraform/Dockerfile
```

3. Update the Terraform Dockerfile to deploy the latest version, as shown in Example 7-81.

Example 7-81 Update Terraform Dockerfile

```
ARG TERRAFORM_VERSION=v1.7.4
```

4. Install Terraform by using the Dockerfile, as shown in Example 7-82.

Example 7-82 Install Terraform

```
[root@y1sprd prd]# cd /root/Terraform  
[root@y1sprd prd]# docker build -t yslterraform .  
[+] Building 300.5s (11/11) FINISHED  
docker:default  
=> [internal] load build definition from Dockerfile 0.0s
```

```

=> => transferring dockerfile: 1.94kB 0.0s
=> [internal] load metadata for docker.mirror.hashicorp.services/alpine:latest
2.2s
=> [internal] load metadata for docker.mirror.hashicorp.services/golang:alpine
2.5s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [builder 1/4] FROM
docker.mirror.hashicorp.services/golang:alpine@sha256:fc5e5848529786cf1136563452b3
3d713d5c60b2c787f6b2a077fa6eeefd9114 8.3s
=> => resolve
docker.mirror.hashicorp.services/golang:alpine@sha256:fc5e5848529786cf1136563452b3
3d713d5c60b2c787f6b2a077fa6eeefd9114 0.0s
=> => sha256:eb8fba61d86413beda3240c40c599041e040e658cd8314e38ee15e67ea57d349
3.24MB / 3.24MB 1.5s
=> => sha256:61981fca6cbfb460db1b5bca98f87e0d4bf4677082bf9a97e229d13fe4656d66
285.19kB / 285.19kB 0.8s
=> => sha256:3b54f06104ceea5cb6d57308b8530d1ad46f64fa50cbcd93e75e79c2dfa17375
68.39MB / 68.39MB 4.3s
=> => sha256:fc5e5848529786cf1136563452b33d713d5c60b2c787f6b2a077fa6eeefd9114
1.65kB / 1.65kB 0.0s
=> =>
sha256:20d1b4c57bf0e1ca0f6f9e7b2b6c6376b55c281730c63d9116162059b04a98101.36kB
/1.36kB 0.0s
=> => sha256:f179f0fe0c7170534c5b43793d57a0f5464042b485e267b247d47cd8dc77133e
2.13kB / 2.13kB 0.0s
=> => sha256:87e02b54b6f2f5aa57c73eef9a1f62166c8eba59be5f0bb100c02d7254dc09e1
174B / 174B 1.1s
=> => sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e68dc38e8acc1 32B
/ 32B 1.4s
=> => extracting
sha256:61981fca6cbfb460db1b5bca98f87e0d4bf4677082bf9a97e229d13fe4656d66 0.0s
=> => extracting
sha256:3b54f06104ceea5cb6d57308b8530d1ad46f64fa50cbcd93e75e79c2dfa17375 2.9s
=> => extracting
sha256:87e02b54b6f2f5aa57c73eef9a1f62166c8eba59be5f0bb100c02d7254dc09e1 0.0s
=> => extracting
sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e68dc38e8acc1 0.0s
=> [stage-1 1/3] FROM
docker.mirror.hashicorp.services/alpine:latest@sha256:c5b1261d6d3e43071626931fc004
f70149baeba2c8ec672bd4f27761f8e1ad6b 1.6s
=> => resolve
docker.mirror.hashicorp.services/alpine:latest@sha256:c5b1261d6d3e43071626931fc004
f70149baeba2c8ec672bd4f27761f8e1ad6b 0.0s
=> => sha256:eb8fba61d86413beda3240c40c599041e040e658cd8314e38ee15e67ea57d349
3.24MB / 3.24MB 1.5s
=> => sha256:c5b1261d6d3e43071626931fc004f70149baeba2c8ec672bd4f27761f8e1ad6b
1.64kB / 1.64kB 0.0s
=> => sha256:5d0da60400afb021f2d8dbfec8b7d26457e77eb8825cba90eba84319133f0efe
528B / 528B 0.0s
=> => sha256:8fc740d8c40e45ea330a3f324fe009148dfc1f771bc90254eaf8ff8bbcecf02
1.47kB / 1.47kB 0.0s
=> => extracting
sha256:eb8fba61d86413beda3240c40c599041e040e658cd8314e38ee15e67ea57d349 0.1s

```

```
=> [builder 2/4] RUN apk add --no-cache git bash openssh      && git clone -b
v1.7.2 https://github.com/hashicorp/terraform.git
/go/src/github.com/hashicorp/terraform 23.0s
=> [builder 3/4] WORKDIR /go/src/github.com/hashicorp/terraform 0.0s
=> [builder 4/4] RUN /bin/bash ./scripts/build.sh 258.3s
=> [stage-1 2/3] COPY --from=builder go/bin /bin 1.3s
=> exporting to image 0.2s
=> => exporting layers 0.2s
=> => writing image
sha256:54d5f158756c1ef22abdda2b053a92f21e279d4f5f40a8361e1d86d9cc0e471c 0.0s
=> => naming to docker.io/library/yslterraform 0.0s
[root@yslprd prd]#
[root@yslprd prd]# docker image list
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
yslterraform	latest	54d5f158756c	7 minutes ago	104MB

5. To verify the Terraform version, use the command shown in Example 7-83.

Example 7-83 Verify Terraform version

```
[root@yslprd ~]# docker run yslterraform version
Terraform v1.7.4
on linux_s390x
```

6. To run the Terraform console, use the command shown in Example 7-84.

Example 7-84 Run Terraform console

```
[root@yslprd prd]# docker run -it --name yslcontainer yslterraform console
```

For more information, see Terraform CLI [documentation](#).

7.10 Deploying Apache Kafka

Apache [Kafka](#) is an open-source distributed event streaming platform, designed to handle high-volume and low-latency data, and that use a distributed architecture with topics, partitions, and brokers to manage data flow across clusters of servers. It enables the publishing, subscribing, storing, and real-time processing of event streams.

The following are the steps to set up Apache Kafka and perform basic operations like creating a topic, writing events to the topic, and reading the events:

1. Install Java OpenJDK 17, as shown in Example 7-85.

Example 7-85 Install Java OpenJDK 17

```
[root@yslprd ~]# sudo yum install -y git java-17-openjdk-devel
```

2. Verify Java version, as shown in Example 7-86.

Example 7-86 Verify Java version

```
[root@yslprd ~]# java -version
openjdk version "17.0.9" 2023-10-17
OpenJDK Runtime Environment Temurin-17.0.9+9 (build 17.0.9+9)
```

OpenJDK 64-Bit Server VM Temurin-17.0.9+9 (build 17.0.9+9, mixed mode, sharing)

3. Create a directory for Kafka, download the Kafka package using **wget**, and then extract it using the **tar** command, as shown in Example 7-87.

Example 7-87 Download Kafka package

```
[root@yslprd ~]# mkdir kafka
[root@yslprd ~]# cd kafka
[root@yslprd kafka]# wget
https://downloads.apache.org/kafka/3.7.0/kafka_2.13-3.7.0.tgz
[root@yslprd kafka]# tar xvf kafka_2.13-3.7.0.tgz
[root@yslprd kafka]# cd kafka_2.13-3.7.0
```

4. Check Kafka version, as shown in Example 7-88.

Example 7-88 Check Kafka version

```
[root@yslprd kafka_2.13-3.7.0]# bin/kafka-topics.sh --version
3.7.0
[root@yslprd kafka_2.13-3.7.0]#
```

5. Kafka relies on ZooKeeper for coordination and management tasks. Start the ZooKeeper service by using the **zookeeper-server-start.sh** script with the specified configuration file, as shown in Example 7-89.

Example 7-89 Start ZooKeeper

```
[root@yslprd kafka_2.13-3.7.0]# bin/zookeeper-server-start.sh
config/zookeeper.properties

[2024-03-25 20:09:07,750] INFO Reading configuration from:
config/zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-03-25 20:09:07,752] WARN config/zookeeper.properties is relative. Prepend ./
to indicate that you're sure!
(org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-03-25 20:09:07,754] INFO clientPortAddress is 0.0.0.0:2181
(org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-03-25 20:09:07,754] INFO secureClientPort is not set
(org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-03-25 20:09:07,754] INFO observerMasterPort is not set
(org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-03-25 20:09:07,754] INFO metricsProvider.className is
org.apache.zookeeper.metrics.impl.DefaultMetricsProvider
(org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-03-25 20:09:07,756] INFO autopurge.snapRetainCount set to 3
(org.apache.zookeeper.server.DataDirCleanupManager)
[2024-03-25 20:09:07,756] INFO autopurge.purgeInterval set to 0
(org.apache.zookeeper.server.DataDirCleanupManager)
[2024-03-25 20:09:07,756] INFO Purge task is not scheduled.
(org.apache.zookeeper.server.DataDirCleanupManager)
[2024-03-25 20:09:07,756] WARN Either no config or no quorum defined in config,
running in standalone mode (org.apache.zookeeper.server.quorum.QuorumPeerMain)
[2024-03-25 20:09:07,758] INFO Log4j 1.2 jmx support not found; jmx disabled.
(org.apache.zookeeper.jmx.ManagedUtil)
(...)
```

```
environment:java.class.path=/root/kafka/kafka_2.13-3.7.0/bin/../libs/activation-1.1.1.jar:/root/kafka/kafka_2.13-3.7.0/bin/../libs/aopalliance-repackaged-2.6.1.jar:/root/kafka/kafka_2.13-3.7.0/bin/../libs/argparse4j-0.7.0.jar:/root/kafka/kafka_2.13-3.7.0/bin/../libs/audience-annotations-0.12.0.jar:/root/kafka/kafka_2.13-3.7.0/bin/
(...)
```

```
[2024-03-26 11:29:41,907] INFO The digest in the snapshot has digest version of 2,
with zxid as 0x1d, and digest value as 58360571118
(org.apache.zookeeper.server.DataTree)
[2024-03-26 11:29:41,919] INFO ZooKeeper audit is disabled.
(org.apache.zookeeper.audit.ZKAuditProvider)
[2024-03-26 11:29:41,921] INFO 1 txns loaded in 4 ms
(org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2024-03-26 11:29:41,921] INFO Snapshot loaded in 31 ms, highest zxid is 0x1e,
digest is 57116772017 (org.apache.zookeeper.server.ZKDatabase)
[2024-03-26 11:29:41,922] INFO Snapshotting: 0x1e to
/tmp/zookeeper/version-2/snapshot.1e
(org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2024-03-26 11:29:41,924] INFO Snapshot taken in 1 ms
(org.apache.zookeeper.server.ZooKeeperServer)
[2024-03-26 11:29:41,930] INFO PrepRequestProcessor (sid:0) started,
reconfigEnabled=false (org.apache.zookeeper.server.PrepRequestProcessor)
[2024-03-26 11:29:41,930] INFO zookeeper.request_throttler.shutdownTimeout = 10000
ms (org.apache.zookeeper.server.RequestThrottler)
[2024-03-26 11:29:41,955] INFO Using checkIntervalMs=60000 maxPerMinute=10000
maxNeverUsedIntervalMs=0 (org.apache.zookeeper.server.ContainerManager)
```

6. Edit the server.properties file to configure the listener settings. Uncomment and update the listeners setting to specify the host name and port for the Kafka broker to listen on and Start the Kafka broker service using the **kafka-server-start.sh** script with the server configuration file. as shown in Example 7-90.

Example 7-90 Start Kafka service

```
[root@yslprd kafka_2.13-3.7.0]# vim /usr/local/etc/kafka/server.properties
```

Here uncomment the server settings and update the value from

```
listeners=PLAINTEXT://:9092
```

to

```
##### Socket Server Settings
```

```
##### The address the socket server listens on. It will
get the value returned from
```

```
# java.net.InetAddress.getCanonicalHostName() if not configured.
```

```
# FORMAT:
```

```
# listeners = listener_name://host_name:port
```

```
# EXAMPLE:
```

```
# listeners = PLAINTEXT://your.host.name:9092
```

```
listeners=PLAINTEXT://localhost:9092
```

```
[root@yslprd kafka_2.13-3.7.0]# bin/kafka-server-start.sh config/server.properties
```

```
[2024-03-26 11:55:45,862] INFO Registered kafka:type=kafka.Log4jController MBean
(kafka.utils.Log4jControllerRegistration$)
```

```
[2024-03-26 11:55:46,026] INFO Setting -D
jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TLS
renegotiation (org.apache.zookeeper.common.X509Util)
[2024-03-26 11:55:46,070] INFO Registered signal handlers for TERM, INT, HUP
(org.apache.kafka.common.utils.LoggingSignalHandler)
[2024-03-26 11:55:46,072] INFO starting (kafka.server.KafkaServer)
[2024-03-26 11:55:46,072] INFO Connecting to zookeeper on localhost:2181
(kafka.server.KafkaServer)
[2024-03-26 11:55:46,090] INFO [ZooKeeperClient Kafka server] Initializing a new
session to localhost:2181. (kafka.zookeeper.ZooKeeperClient)
[2024-03-26 11:55:46,156] INFO Client
environment:zookeeper.version=3.8.3-6ad6d364c7c0bcf0de452d54ebefa3058098ab56,
built on 2023-10-05 10:34 UTC (org.apache.zookeeper.ZooKeeper)
[2024-03-26 11:55:46,156] INFO Client environment:host.name=yslprd
(org.apache.zookeeper.ZooKeeper)
[2024-03-26 11:55:46,157] INFO Client environment:java.version=17.0.9
(org.apache.zookeeper.ZooKeeper)
[2024-03-26 11:55:46,157] INFO Client environment:java.vendor=Eclipse Adoptium
(org.apache.zookeeper.ZooKeeper)
[2024-03-26 11:55:46,157] INFO Client environment:java.home=/opt/java/jdk
(org.apache.zookeeper.ZooKeeper)
[2024-03-26 11:55:46,157] INFO Client
environment:java.class.path=/root/kafka/kafka_2.13-3.7.0/bin/../libs/activation-1.
1.1.jar:/root/kafka/kafka_2.13-3.7.0/bin/../libs/aopalliance-repackaged-2.6.1.jar:/
root/kafka/kafka_2.13-3.7.0/bin/../libs/argparse4j-0.7.0.jar:/root/kafka/kafka_2.
13-3.7.0/bin/../libs/audience-annotations-0.12.0.jar:/root/kafka/kafka_2.13-3.7.0/
bin/../libs/caffeine-2.9.3.jar:/root/kafka/kafka_2.13-3.7.0/bin/../libs/checker-qu
al-3.19.0.jar:/root/kafka/kafka_2.13-3.7.0/bin/../libs/commons-beanutils-1.9.4.jar
:/root/kafka/kafka_2.13-3.7.0/bin/../libs/commons-cli-1.4.jar:/root/kafka/kafka_2.
13-3.7.0/bin/../libs/commons-collections-3.2.2.jar:/root/kafka/kafka_2.13-3.7.0/bi
n/../libs/commons-digester-2.1.jar:/root/kafka/kafka_2.13-3.7.0/bin/../libs/common
s-io-2.11.0.jar:/root/kafka/kafka_2.13-3.7.0/bin/../libs/commons-lang3-3.8.1.jar:

(...)

```

```
[2024-03-26 11:55:46,991] WARN [Controller id=0, targetBrokerId=0] Connection to
node 0 (localhost/127.0.0.1:9092) could not be established. Node may not be
available. (org.apache.kafka.clients.NetworkClient)
[2024-03-26 11:55:47,013] INFO [Controller id=0, targetBrokerId=0] Client
requested connection close from node 0 (org.apache.kafka.clients.NetworkClient)
[2024-03-26 11:55:47,030] INFO [ExpirationReaper-0-AlterAcls]: Starting
(kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2024-03-26 11:55:47,061] INFO [/config/changes-event-process-thread]: Starting
(kafka.common.ZkNodeChangeNotificationListener$ChangeEventProcessThread)
[2024-03-26 11:55:47,065] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0]
Enabling request processing. (kafka.network.SocketServer)
[2024-03-26 11:55:47,068] INFO Awaiting socket connections on localhost:9092.
(kafka.network.DataPlaneAcceptor)
[2024-03-26 11:55:47,084] INFO Kafka version: 3.7.0
(org.apache.kafka.common.utils.AppInfoParser)
[2024-03-26 11:55:47,084] INFO Kafka commitId: 2ae524ed625438c5
(org.apache.kafka.common.utils.AppInfoParser)

```

```
[2024-03-26 11:55:47,084] INFO Kafka startTimeMs: 1711472147080
(org.apache.kafka.common.utils.AppInfoParser)
[2024-03-26 11:55:47,088] INFO [KafkaServer id=0] started
(kafka.server.KafkaServer)
[2024-03-26 11:55:47,204] INFO
[zk-broker-0-to-controller-alter-partition-channel-manager]: Recorded new
controller, from now on will use node localhost:9092 (id: 0 rack: null)
(kafka.server.NodeToControllerRequestThread)
[2024-03-26 11:55:47,225] INFO
[zk-broker-0-to-controller-forwarding-channel-manager]: Recorded new controller,
from now on will use node localhost:9092 (id: 0 rack: null)
(kafka.server.NodeToControllerRequestThread)
```

7. Use the **kafka-topics.sh** script to create a Kafka topic named "Redbooks-SG248217-events" using the **--create** option and specifying the bootstrap server (localhost:9092), as shown in Example 7-91.

Example 7-91 Create topic

```
[root@yslprd kafka_2.13-3.7.0]# bin/kafka-topics.sh --create --topic
Redbooks-SG248217-events --bootstrap-server localhost:9092
Created topic Redbooks-SG248217-events.
[root@yslprd kafka_2.13-3.7.0]#
```

8. After creating the topic, describe its details using the **kafka-topics.sh** script with the **--describe** option and specifying the topic name and bootstrap server. This command provides information about the topic, including its partitions, replication factor, and configuration, as shown in Example 7-92.

Example 7-92 Describe topic

```
[root@yslprd kafka_2.13-3.7.0]# bin/kafka-topics.sh --describe --topic
Redbooks-SG248217-events --bootstrap-server localhost:9092
Topic: Redbooks-SG248217-eventsTopicId: 7RRRA1Y3SuKMUTDMpnPrPAPartitionCount: 1
ReplicationFactor: 1Configs:
    Topic: Redbooks-SG248217-eventsPartition: 0Leader: 0Replicas: 0Isr: 0
[root@yslprd kafka_2.13-3.7.0]#
```

9. Use the **kafka-console-producer.sh** script to write events/messages to the Kafka topic "Redbooks-SG248217-events". This interactive console allows you to input event messages that will be produced to the specified topic, as shown in Example 7-93.

Example 7-93 Write events to the Kafka topic

```
[root@yslprd kafka_2.13-3.7.0]# bin/kafka-console-producer.sh --topic
Redbooks-SG248217-events --bootstrap-server localhost:9092
>Redbooks-SG248217 Write Chapter 1 Event
>Redbooks-SG248217 Add Figure 1 Event
>Redbooks-SG248217 Add Figure 2 Event
>
```

10. Finally, use the **kafka-console-consumer.sh** script to consume and read events/messages from the Kafka topic "Redbooks-SG248217-events". Specify the topic name, start reading from the beginning of the topic (**--from-beginning**), and provide the bootstrap server information, as shown in Example 7-94 on page 216.

Example 7-94 Read events from the Kafka topic

```
[root@yslprd kafka_2.13-3.7.0]# bin/kafka-console-consumer.sh --topic
Redbooks-SG248217-events --from-beginning --bootstrap-server localhost:9092
Redbooks-SG248217 Write Chapter 1 Event
Redbooks-SG248217 Add Figure 1 Event
Redbooks-SG248217 Add Figure 2 Event
```

For more information, see [Apache Kafka](#).

7.11 Deploying Validated Open Source Software

Many open-source packages have been successfully ported and/or validated on corresponding distribution versions by IBM. For more detailed information, see [Validated Open Source Software](#).

Figure 7-21 shows the latest data on open-source packages that have been adapted or confirmed for compatibility with corresponding distribution versions by IBM. This information includes links to packaged binaries or documentation for building them on Linux on IBM Z. Additionally, the Dockerfile/Image column provides links to Dockerfile or docker images for certain packages.

IBM Community							
IBM Z and LinuxONE Community Participate Topic groups User groups Solutions Events Resources							
Packages	RHEL 9.x	Ubuntu 22.x	SLES 15.x	Dockerfile/Image	RHEL 8.x/7.x	Ubuntu 20.x	SLES 12.x
Alfresco	NA	7.x	NA	Via 7.x	7.x	7.x	NA
Ansible	Distro Latest	Distro Latest	Distro Latest	NA	Distro Latest	Distro Latest	Distro Latest
AntLR	4.x	Distro 4.x	4.x	4.x	4.x	Distro 4.x	4.x
Apache ActiveMQ	Latest	Distro Latest	Latest	NA	Latest	Distro Latest	Latest
Apache Camel	Latest	Latest	Latest	NA	Latest	Latest	Latest
Apache Cassandra	4.x	4.x	3.x, 4.x	Image	2.x, 3.x, 4.x	3.x, 4.x	2.x, 3.x, 4.x
Apache CouchDB	Latest	Latest	NA	Image	3.x	Latest	NA
Apache Flume	Download	Download	Download	NA	Download	Download	Download
Apache Geode	Latest	Latest	Latest	1.x	Latest	Latest	Latest
Apache HBase	2.x	2.x	2.x	NA	2.x	2.x	2.x
Apache HTTP	Distro 2.4	Distro 2.4	Distro 2.4	Image	Distro 2.4	Distro 2.4	Distro 2.4
Apache Ignite	Latest	Latest	Latest	Image	Latest	Latest	Latest
Apache JMeter	Latest	Distro Latest	Latest	NA	Latest	Distro Latest	Latest
Apache Kafka	Latest	Latest	Latest	NA	Latest	Latest	Latest

Figure 7-21 Validated Open Source Software on IBM Linux on IBM Z

Note: Note that binaries or Docker images provided by the community may not be regularly validated. For further details, please consult the official documentation.

7.12 Containerized Workloads

Running applications as containers has become prominent for cloud-native developed software and solutions. There is also a trend in most companies to modernize their traditional applications, which implies, among other methods and techniques, decoupling monolithic applications into microservices and running them as containers on a container orchestrating platform.

Containerization is the process of packaging compiled software code, libraries, configuration, compute, network, and storage definitions into an immutable image, which can be deployed as a set of containers and managed by a container management tool, allowing automation, flexible scalability, and operations. By sharing the host kernel with the running instances, containerization provides a lightweight solution, avoiding overhead of virtualization.

For more information on this topic, see [Containers versus virtual machines \(VMs\): What's the difference?](#).

7.12.1 Solutions for container-based workloads on IBM Z

Containers are deployed on a runtime engine, such as Docker Engine, Open Containers Initiative (OCI), Ubuntu Linux Containers (LXC), Containerd, among others.

For leveraging production environments, the scalability, mainframe, and cluster management are provided by some container orchestration tools, such as:

- ▶ Kubernetes
- ▶ Red Hat OpenShift Container Platform
- ▶ Docker Swarm
- ▶ Rancher
- ▶ Ubuntu Linux Container Daemon (LXD)

All of the above are market solutions available for x86 and also compatible with and supported on Linux on IBM Z platform (s390x).

Additionally, IBM Z has solutions designed to run on the z/OS operating system. They are:

- ▶ **IBM z/OS Container Extensions (zCX):** IBM z/OS Container Extensions (zCX) is a z/OS 2.4+ feature that enables clients to deploy Linux applications as Docker containers on z/OS as part of a z/OS workload. This maintains operational control of the Linux environment within z/OS, brings z/OS qualities of service to the application deployment, and does not require the provisioning of separate LPARs or system images.
- ▶ **IBM zCX Foundation for Red Hat OpenShift:** IBM zCX Foundation for Red Hat OpenShift (zCX for OpenShift) allows for an agile and flexible deployment of Linux on Z applications and software in a self-contained Red Hat OpenShift cluster into IBM z/OS. Co-locate applications and workloads while simultaneously taking advantage of the many z/OS Qualities of Service (QoS). Developers can integrate Linux on Z applications with z/OS to deliver enterprise-level container orchestration and management capabilities around containerized software.
- ▶ **IBM z/OS Container Platform (zOSCP):** The IBM z/OS Container Platform is designed to enable users to realize the benefits of a cloud native development strategy within z/OS system. Run containers natively on IBM z/OS gaining the benefits of industry-standard open source container technologies while also taking advantage of the security, reliability and performance benefits of IBM Z.

7.12.2 Running a container on Linux on IBM Z

In this section we outline the steps to show how to run a container on Linux on IBM Z and aspects to consider when migrating from x86 to Linux on IBM Z.

1. Check podman version. Example 7-95 provides an example of the command we used to check our podman version on IBM Z.

Example 7-95 Checking podman version

```
[root@rhel93 ~]# podman version
Client:      Podman Engine
Version:     4.6.1
API Version: 4.6.1
Go Version:  go1.20.6
Built:       Fri Aug 25 06:07:59 2023
OS/Arch:     linux/s390x
```

2. Pull the image from a container registry (quay.io/podman) and run the hello-world container by using podman, as shown in Example 7-96. We are running the hello-world container with podman on Red Hat Enterprise Linux 9.3 on IBM Z.

Example 7-96 Pull image and run container

```
[root@rhel93 ~]# podman run hello-world
Resolved "hello-world" as an alias
(/etc/containers/registries.conf.d/000-shortnames.conf)
Trying to pull quay.io/podman/hello:latest...
Getting image source signatures
Copying blob 7d2fe9f4362b done
Copying config 6579165753 done
Writing manifest to image destination
```

```
!... Hello Podman World ...!
```

```

      .--"---.
     /  -   - \
    / (0)   (0) \
   ~~~|  =(,Y,)=  |
      .---. /~ \ |~~
 ~/  o  o \~~~~~.---. ~~
 |  =(X)=  |~ / (0 (0) \
 ~~~~~~ ~ ~|  =(Y_)=  |
 ~~~~~ ~~~~|  U      |~~
```

```
Project:  https://github.com/containers/podman
Website:  https://podman.io
Desktop:  https://podman-desktop.io
Documents: https://docs.podman.io
YouTube:  https://youtube.com/@Podman
X/Twitter: @Podman_io
Mastodon: @Podman_io@fosstodon.org
```

3. List the image, hello-world, by running the command shown in Example 7-97 on page 219.

Example 7-97 List the image

```
[root@rhel193 ~]# podman image ls hello-world
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
quay.io/podman/hello	latest	65791657534c	22 hours ago	1.43 MB

4. Inspect the image and manifests. Example 7-98 shows the command used to inspect the image and the manifests for the image hello-world as well as its output..

Example 7-98 Inspect the image

```
[root@rhel193 ~]# podman image inspect hello-world
```

```
[
  {
    "Id":
      "65791657534c4d987bf940349ae62866f45cca0ba64fedaa3f5f55bddab57cf7",
    "Digest":
      "sha256:302fce3ce63131ab2cdc650041362cce505743258eda1df6dd61274676038ed4",
    "RepoTags": [
      "quay.io/podman/hello:latest"
    ],
    "RepoDigests": [

      "quay.io/podman/hello@sha256:302fce3ce63131ab2cdc650041362cce505743258eda1df6dd61274676038ed4",

      "quay.io/podman/hello@sha256:e0e993a5330b5f76dcc3ed46ae824c7c57f62973677cffc4bd
      ae71ba5301edc3"
    ],
    "Parent": "",
    "Comment": "",
    "Created": "2024-03-10T02:15:25.819866618Z",
    "Config": {
      "Env": [

        "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
      ],
      "Cmd": [
        "/usr/local/bin/podman_hello_world"
      ],
      "Labels": {
        "artist": "Máirín Ní ?u??ai?, X/Twitter:@mairin",
        "io.buildah.version": "1.23.1",
        "io.containers.capabilities": "sys_chroot",
        "maintainer": "Podman Maintainers",
        "org.opencontainers.image.description": "Hello world image
        with ascii art",
        "org.opencontainers.image.documentation":
        "https://github.com/containers/PodmanHello/blob/a804b8b30e8fc399c270d83a2596778
        9e81ca659/README.md",
        "org.opencontainers.image.revision":
        "a804b8b30e8fc399c270d83a25967789e81ca659",
        "org.opencontainers.image.source":
        "https://raw.githubusercontent.com/containers/PodmanHello/a804b8b30e8fc399c270d
        83a25967789e81ca659/Containerfile",
        "org.opencontainers.image.title": "hello image",
```

```

        "org.opencontainers.image.url":
"https://github.com/containers/PodmanHello/actions/runs/8218879622"
    },
    "Version": "",
    "Author": "",
    "Architecture": "s390x",
    "Os": "linux",
    "Size": 1430025,
    "VirtualSize": 1430025,
    "GraphDriver": {
        "Name": "overlay",
        "Data": {
            "UpperDir":
"/var/lib/containers/storage/overlay/4707b5123995ac03b8f4c37937fa34055e668691be
6c97505dd097f328a8225e/diff",
            "WorkDir":
"/var/lib/containers/storage/overlay/4707b5123995ac03b8f4c37937fa34055e668691be
6c97505dd097f328a8225e/work"
        }
    },
    "RootFS": {
        "Type": "layers",
        "Layers": [

"sha256:4707b5123995ac03b8f4c37937fa34055e668691be6c97505dd097f328a8225e"
        ]
    },
    "Labels": {
        "artist": "Máirín Ní ?u??ai?, X/Twitter:@mairin",
        "io.buildah.version": "1.23.1",
        "io.containers.capabilities": "sys_chroot",
        "maintainer": "Podman Maintainers",
        "org.opencontainers.image.description": "Hello world image with
ascii art",
        "org.opencontainers.image.documentation":
"https://github.com/containers/PodmanHello/blob/a804b8b30e8fc399c270d83a2596778
9e81ca659/README.md",
        "org.opencontainers.image.revision":
"a804b8b30e8fc399c270d83a25967789e81ca659",
        "org.opencontainers.image.source":
"https://raw.githubusercontent.com/containers/PodmanHello/a804b8b30e8fc399c270d
83a25967789e81ca659/Containerfile",
        "org.opencontainers.image.title": "hello image",
        "org.opencontainers.image.url":
"https://github.com/containers/PodmanHello/actions/runs/8218879622"
    },
    "Annotations": {},
    "ManifestType":
"application/vnd.docker.distribution.manifest.v2+json",
    "User": "",
    "History": [
        {
            "created": "2024-03-10T02:15:25.712953941Z",

```

```

        "created_by": "/bin/sh -c #(nop) LABEL maintainer=\"Podman
Maintainers\"",
        "empty_layer": true
    },
    {
        "created": "2024-03-10T02:15:25.712965463Z",
        "created_by": "/bin/sh -c #(nop) LABEL artist=\"Máirín Ní
?u??ai?, X/Twitter:@mairin\"",
        "empty_layer": true
    },
    {
        "created": "2024-03-10T02:15:25.712986973Z",
        "created_by": "/bin/sh -c #(nop) LABEL
io.containers.capabilities=\"sys_chroot\"",
        "empty_layer": true
    },
    {
        "created": "2024-03-10T02:15:25.819305094Z",
        "created_by": "/bin/sh -c #(nop) COPY
file:1e42fe12ca1964e805d3964cf2091f193b0a612c7f16e4162446b29882ffc3af in
/usr/local/bin/podman_hello_world ",
        "empty_layer": true
    },
    {
        "created": "2024-03-10T02:15:25.825912222Z",
        "created_by": "/bin/sh -c #(nop) CMD
[\"/usr/local/bin/podman_hello_world\"]"
    }
],
    "NamesHistory": [
        "quay.io/podman/hello:latest"
    ]
}
]

```

Note: The image is offered by the registry as a manifest for the architecture IBM Z (s390x).

5. Inspect the image multi-architecture manifests by using the command shown in Example 7-99.

Example 7-99 Inspecting the image multi-architecture manifests

```

[root@rhel93 ~]# podman manifest inspect hello-world
{
    "schemaVersion": 2,
    "mediaType": "application/vnd.docker.distribution.manifest.list.v2+json",
    "manifests": [
        {
            "mediaType":
"application/vnd.docker.distribution.manifest.v2+json",
            "size": 427,
            "digest":
"sha256:3837a62d01fa5342d4756091ee4795593cf3742f93bdfc4a54a793e88c5a1f92",

```

```

        "platform": {
            "architecture": "amd64",
            "os": "linux"
        }
    },
    {
        "mediaType":
"application/vnd.docker.distribution.manifest.v2+json",
        "size": 427,
        "digest":
"sha256:5e7bc12b54286f20dbf75e674b579c3757fd176df059557c1cb8fe720498c0ee",
        "platform": {
            "architecture": "arm64",
            "os": "linux"
        }
    },
    {
        "mediaType":
"application/vnd.docker.distribution.manifest.v2+json",
        "size": 427,
        "digest":
"sha256:f2beb2924f83784b789f1d48ee6c6de91ec81a178f1ff158ac3605696be0dbb6",
        "platform": {
            "architecture": "ppc64le",
            "os": "linux"
        }
    },
    {
        "mediaType":
"application/vnd.docker.distribution.manifest.v2+json",
        "size": 427,
        "digest":
"sha256:e0e993a5330b5f76dcc3ed46ae824c7c57f62973677cffc4bdae71ba5301edc3",
        "platform": {
            "architecture": "s390x",
            "os": "linux"
        }
    }
]
}

```

As we could verify, to run a container on IBM Z, it is necessary to have an image built for the s390x architecture.

Most of the popular and highly used open-source software are available for IBM Z architecture on the main public container image registries, for instance at:

- ▶ [Docker Hub](#)
- ▶ [Red Hat Quay.io](#)
- ▶ [IBM Cloud Container Registry](#)

7.12.3 Building a container for Linux on IBM Z

There are several ways to build an image for different platforms., For the scope of this book, we will demonstrate a build process of an image for IBM Z (s390x) architecture.

In this section, we demonstrates the commands and their results when building an nginx static web page on Red Hat Enterprise Linux 9.3 on IBM Z by using the Docker runtime.

Example 7-100 provides the command and its result when verifying a Docker runtime version.

Example 7-100 Verifying docker runtime version

```
[root@rhel93 ~]# docker version
Client: Docker Engine - Community
 Version:           25.0.3
 API version:       1.44
 Go version:        go1.21.6
 Git commit:        4debf41
 Built:             Fri Feb  9 07:00:46 2024
 OS/Arch:           linux/s390x
 Context:           default

Server: Docker Engine - Community
 Engine:
  Version:           25.0.3
  API version:       1.44 (minimum version 1.24)
  Go version:        go1.21.6
  Git commit:        f417435
  Built:             Fri Feb  9 06:57:18 2024
  OS/Arch:           linux/s390x
  Experimental:      false
 containerd:
  Version:           1.6.28
  GitCommit:         ae07eda36dd25f8a1b98dfbf587313b99c0190bb
 runc:
  Version:           1.1.12
  GitCommit:         v1.1.12-0-g51d5e94
 docker-init:
  Version:           0.19.0
  GitCommit:         de40ad0
```

Example 7-101 shows our HTML source code of the web page that we will deploy on top of the nginx server.

Example 7-101 HTML source

```
[root@rhel93 ~]# cat html/index.html
<!doctype html>
<html>
  <head>
    <title>Practical migration from x86 to Linux on IBM Z</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>
      Practical migration from x86 to Linux on IBM Z
```

```
</h1>
</body>
</html>
```

By using the command shown in Example 7-102, we pulled the nginx image from the Docker Hub repository (docker.io/library), which will be used as the base image of our application.

Example 7-102 Pull the nginx image

```
[root@rhel93 ~]# docker pull nginx:latest
latest: Pulling from library/nginx
e55f0b78e9a1: Already exists
00d71c6ab3fd: Already exists
eb532096aa4e: Already exists
3dca1ddb5d22: Already exists
c094e4fa325e: Already exists
4db065638e5d: Already exists
fc4bfcfa61e6: Already exists
Digest: sha256:c26ae7472d624ba1fafd296e73cecc4f93f853088e6a9c13c0d52f6ca5865107
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

We then list and review the nginx image and verify that it is compatible with IBM Z (Example 7-103).

Example 7-103 List and review

```
[root@rhel93 ~]# docker image ls nginx
REPOSITORY TAG IMAGE ID CREATED SIZE
nginx latest 1c3941931a19 3 weeks ago 165MB
```

```
[root@rhel93 ~]# docker image inspect nginx |grep Arch
"Architecture": "s390x",
```

To build the application, we will prepare a Dockerfile (Containerfile for OCI), which specifies the configuration for the subsequent image. Example 7-104 shows our Dockerfile specifies nginx as the base image and adds the web page file within the html nginx directory.

Example 7-104 Preparing the Dockerfile

```
[root@rhel93 ~]# cat Dockerfile
FROM nginx:latest
ADD html /usr/share/nginx/html
```

Example 7-105 shows the command to build the image with the name redbook-container:s390x from the Dockerfile created in Example 7-104.

Example 7-105 Build the image

```
[root@rhel93 ~]# docker build -t redbook-container:s390x .
[+] Building 0.5s (8/8) FINISHED
docker:default
=>[internal] load build definition from Dockerfile
0.0s
```



```

=>=> transferring dockerfile: 147B
0.0s
=> [internal] load metadata for docker.io/library/nginx:latest
0.5s
=> [auth] library/nginx:pull token for registry-1.docker.io
0.0s
=> [internal] load .dockerignore
0.0s
=>=> transferring context: 2B
0.0s
=> [internal] load build context
0.0s
=>=> transferring context: 181B
0.0s
=> [1/2] FROM
docker.io/library/nginx:latest@sha256:c26ae7472d624ba1fafd296e73cecc4f93f853088e6a
9c13c0d52f6ca5865107
0.0s
=> => resolve
docker.io/library/nginx:latest@sha256:c26ae7472d624ba1fafd296e73cecc4f93f853088e6a
9c13c0d52f6ca5865107
0.0s
=> CACHED [2/2] ADD html /usr/share/nginx/html
0.0s
=> exporting to image
0.0s
=>=> exporting layers
0.0s
=> => writing image
sha256:06bcb794c39a7923ed7864c639a57ae3b57202ae4e09e90c751adeea0c5fa760
0.0s
=>=> naming to docker.io/library/redbook-container:s390x
0.0s

```

The command shown in Example 7-106 allows you to list and inspect the redbook-container image.

Example 7-106 List and inspect the container image

```

[root@rhel93 ~]# docker image ls redbook-container

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
redbook-container	s390x	06bcb794c39a	6 minutes ago	165MB

```

[root@rhel93 ~]# docker inspect redbook-container:s390x |grep Arch
"Architecture": "s390x",

```

Next, we use the command shown in Example 7-107 to run a container from the image that redbook-container:s390x that we built. Nginx exposes port 80 TCP, then we mapped the port 8080 TCP on the host to the port 80 of the container.

Example 7-107 Run the container

```

[root@rhel93 ~]# docker run --name redbook-container -d -p 8080:80
redbook-container:s390x

```

67a6c4b1e56165f1cf23557bfe352099e92c6ca956b8954ac23638d1b78227ca

To validate that the container is running, use the command shown in Example 7-108.

Example 7-108 Validate running container

```
[root@rhel93 ~]# docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS		
NAMES			
67a6c4b1e561	redbook-container:s390x	"/docker-entrypoint...."	40 seconds ago
Up 39 seconds	0.0.0.0:8080->80/tcp, :::8080->80/tcp		
redbook-container			

Figure 7-22 shows the resulting web page.



Figure 7-22 web application - nginx



Hands-on migration

Now that the migration planning and analysis tasks are completed, test your migration plan by performing a hands-on migration in a test environment. During this process, load test the environment and ensure that the performance meets business expectations and needs. After performing the test migration, plan for the migration of the production workloads.

Many methods of migrating your data from your source x86 servers to Linux on IBM Z are available, and many different ways of configuring your new IBM Z environment exist. A typical migration plan often involves a deep architectural understanding of the source application, as described in Chapter 6, “Migration analysis” on page 71.

After the main migration tasks are completed, applications can be tuned to use many of the platform features, such as the Pause-less Garbage Collection for Java workloads, leveraging the IBM Z’s Guarded Storage Facility for improved throughput and response times, and Pervasive Encryption, for encryption of data in-flight and at-rest.

This chapter describes a hands-on migration scenario that is performed in our lab in which we migrated a full working application, which was composed of a WebSphere Application Server ND cluster and Db2, from several x86 servers to Linux on IBM Z. We also illustrate the migration process from an WebSphere Application Server Liberty/Open Liberty instance on x86 to Linux on IBM Z.

This chapter includes the following sections:

- ▶ 8.1, “Setting up the system” on page 228
- ▶ 8.2, “Migrating Db2 and its data” on page 229
- ▶ 8.3, “Migrating WebSphere Application Server Network Deployment cluster” on page 233
- ▶ 8.4, “Migrating WebSphere Application Server Liberty and Open Liberty” on page 238
- ▶ 8.5, “Migrating Fibre Channel devices” on page 243

8.1 Setting up the system

This section describes the required tasks to create virtual guests on Linux on IBM Z. We first analyzed our reference application architecture to determine what resources were being used in the x86 environment. During that process, we completed the planning checklists regarding that architecture. Next, we determined what to use in the IBM Z environment, keeping performance in mind. Finally, we performed the migration tasks.

Our application represents most of the workloads that used by the organizations. In that sense, it is worth mentioning that more simpler or complex scenarios might exist, which can require different migration approaches in accordance to each application and businesses requirements.

8.1.1 Software products and tools worksheets

During the meetings with our key stakeholders, it was decided that all of the components that are used by our reference application must be migrated to Linux on IBM Z. It was also decided that the same levels of each software technology that is used in the source x86 workloads must be in place under our target Linux on IBM Z guests. This migration approach, on which the same levels of software are implemented in the target system while retaining the same application architecture, is often referred to “as-is”. Table 8-1 lists the software products that compose our source x86 application.

Table 8-1 Software products and tools checklist for the x86 environment

Software products and tools worksheet for x86 environment				
Name	Version	Vendor/Source website	License type	Linux on IBM Z
Db2	11.5.4.0	IBM www.ibm.com	Proprietary	Supported
WebSphere Application Server ND	9.0.5.5	IBM www.ibm.com	Proprietary	Supported

8.1.2 Hardware worksheet

This section lists the physical and virtual hardware resources that are needed. For the servers that are used in this project, the source environment’s hardware resources were gathered and those resources were used as a reference before provisioning our Linux on IBM Z guests. Table 8-2 shows one of the checklists that were used to map the source x86 workloads to our target Linux on IBM Z guests.

Table 8-2 Hardware worksheet for the x86 environment

HARDWARE PLANNING WORKSHEET			
SERVERNAME: rdbk86r1			
RESOURCE	Source (x86)	Destination (Linux on IBM Z)	Observation
Number of CPU	8	1	Virtual to Virtual
System memory (in GB)	32	16	
OS SWAP Memory (in GB)	4	.8	

HARDWARE PLANNING WORKSHEET			
SERVERNAME: rdbk86r1			
Network connection^a			
Connection Description	Gigabit Ethernet	Gigabit Ethernet	
Connection Type	Gigabit Ethernet	Gigabit Ethernet	
IP Address/Netmask	129.40.23.229/28	129.40.23.231/28	
Vlan number: vSwitch	2	2 : vSwitch1	
Disk Resource^b			
OS File system	LVM	LVM	Root
Mount Point: Size (in GB): Type		/root	Logical Volume
Volume Groups (VG): vg_system VG: 40 GB vg_websphere VG: 40 GB vg_data VG: 10 GB			

- a. For IBM Z, the available network connections are:
- QETH
 - HiperSockets
 - Direct OSA-Express connection
- b. We used the Logical Volume Manager (LVM) for the Linux environment since it provides flexibility and reduces downtime of the environment with online resizing of the logical volumes

8.2 Migrating Db2 and its data

The migration of Db2 and other DBMS systems involves input from the key business stakeholders, the database administrator, and the primary systems administrators from the source x86 and target Linux on IBM Z system. The migration strategy that is used typically depends on several factors, such as the size of the databases, number of databases to be migrated, network speed, storage and backup technologies that are used, and back-out plans.

Migrating databases typically involve a downtime, specially when handling production data. Always ensure that the source Db2 instances are stopped and that the applications that rely on it are routed to the new server right after the migration is complete. Failure to do so can result in the target system running out of sync in relation to its source server in such a way that the migration steps must be repeated from scratch.

Note: Always perform a full database backup before the migration occurs. Taking this precaution prevents common human errors during the process and also allows for a point-in-time database restore (if necessary).

This section describes the migration of Db2 data from a source x86 system (1nx-x86-db2) to a Linux on IBM Z guest (1nx-z-db2). We perform the migration of the same database by using two different methods: the db2move/db2look and the LOAD FROM CURSOR utilities. To perform these steps, you should use the Db2 administrator user ID.

8.2.1 Preliminary migration steps

From within the source x86 system, switch to the Db2 instance ID (db2inst1 in our environment) and perform a full database backup, as shown in Example 8-1. Notice how we ensure that the database is not in use before proceeding with its backup.

Example 8-1 Perform a full database backup before any other activity occurs

```
lnx-x86-db2 $ db2 list applications
SQL1611W  No data was returned by Database System Monitor.
lnx-x86-db2 $ db2 deactivate db SAMPLE
DB20000I  The DEACTIVATE DATABASE command completed successfully.
lnx-x86-db2 $ db2 BACKUP DATABASE SAMPLE TO /db2_backup COMPRESS
```

Backup successful. The timestamp for this backup image is : 20240322141859

```
lnx-x86-db2 $ ls -ltr /db2_backup/
total 28768
-rw----- 1 db2inst1 db2admin 29458432 Mar 22 14:19
SAMPLE.0.db2inst1.DBPART000.20240322141859.001
```

Still under the source x86 systems, retrieve the list of authorization IDs that are required for your application to function, as shown in Example 8-2. In our application, we are required to have a group that is named appdb created. The members of this group are granted privileges to our migrated databases. See the Db2 product documentation and your stakeholders for more information about requirements, depending on your environment.

Example 8-2 Retrieve the list of grantees for our database

```
lnx-x86-db2 $ db2 connect to SAMPLE
```

Database Connection Information

```
Database server      = DB2/LINUX8664 11.5.4.0
SQL authorization ID = DB2INST1
Local database alias = SAMPLE
```

```
lnx-x86-db2 $ db2 select '*' from SYSIBMADM.AUTHORIZATIONIDS
```

```
AUTHID AUTHIDTYPE
-----
APPDB G
PUBLIC G
SYSDEBUG R
SYSDEBUGPRIVATE R
SYSTS_ADM R
SYSTS_MGR R
DB2INST1 U
```

7 record(s) selected.

From the target Linux on IBM Z, create the Db2 instance ID that houses your databases. It is a good practice to also create a dedicated common group for all instances you are going to use. Also, create the authorization IDs and groups that are required for your application to function. Example 8-3 on page 231 shows the creation of our required application Db2 IDs and groups.

Example 8-3 Required ID creation under lnx-z-db2

```
lnx-z-db2 # groupadd appdb
lnx-z-db2 # useradd -g appdb -c 'User required by our application' -m appuser
lnx-z-db2 # groupadd db2admin
lnx-z-db2 # useradd -g db2admin -c 'Db2 instance ID' -m db2inst1
lnx-z-db2 # useradd -g db2admin -c 'Db2 fenced ID' -m db2fenc1
```

After the Db2 product installation under the lnx-z-db2 server, we created our application required file systems and adjusted the required kernel parameters. Then, we created the Db2 instance. Example 8-4 shows the creation of the db2inst1 Enterprise Server Edition (ese) instance that listens on TCP port 50000 and uses db2fenc1 as its fenced ID.

Example 8-4 Db2 instance creation

```
lnx-z-db2 # /opt/ibm/db2/V11.5/instance # ./db2icrt -s ese -p 50000 -u db2fenc1
db2inst1
DBI1446I The db2icrt command is running.
(... output suppressed ...)
The execution completed successfully.
```

For more information, see the DB2 installation log at "/tmp/db2icrt.log.3952".
DBI1070I Program db2icrt completed successfully.

Finally, start the previously created Db2 instance so that it can be used during our migration, as shown in Example 8-5.

Example 8-5 Start the Db2 instance

```
lnx-z-db2 # su - db2inst1
lnx-z-db2 $ db2start
03/22/2024 14:29:12 0 0 SQL1063N DB2START processing was successful.
SQL1063N DB2START processing was successful.
```

8.2.2 Data migration using db2move and db2look

Use the db2look tool to extract the required Data Definition Language (DDL) statements to reproduce the database objects of one database into another database. The tool can also generate the required SQL statements to replicate the statistics from the one database to the other, and the statements that are needed to replicate the database configuration, database manager configuration, and registry variables. Notice that the **db2look** command does not extract any table data. This is described later in this section and uses the db2move tool.

Example 8-6 shows how to use the **db2look** command to generate the required DDL statements for replicating our source x86 database objects to our target Linux on IBM Z system.

Example 8-6 Generate the DDL by using the DB2LOOK command

```
lnx-x86-db2 $ db2look -d SAMPLE -e -x -l -createdb -o /db2_temp/db2look/SAMPLE.sql
-- No userid was specified, db2look tries to use Environment variable USER
-- USER is: DB2INST1
-- Creating DDL for table(s)
-- Output is sent to file: /db2_temp/db2look/SAMPLE.sql
```

The command includes the following parameters:

- ▶ -d: Name of database
- ▶ -e: Extract the database objects
- ▶ -x: Generates authorization DDL statements such as GRANT statements.
- ▶ -l: Generates DDL statements for user-defined database objects
- ▶ -createdb: Generates the **CREATE DATABASE** command that was used to create the source database.
- ▶ -o: The name of the output file

For more information about the db2look tool, see the IBM Db2 product documentation that is available at [IBM Knowledge Center](#).

The Db2 Backup resulting file cannot be used to move data between x86 and Linux on IBM Z operating systems. Use the db2move utility to export the source x86 tables data, as shown in Example 8-7. By default, the db2move utility generates its exported files in the current working directory. Ensure that you switch to the wanted destination directory before running this command because several files are created for every table in the database.

Example 8-7 db2move export utility

```
lnx-x86-db2 $ cd /db2_temp/db2move
lnx-x86-db2 $ db2move SAMPLE export
lnx-x86-db2 $ ls
db2move.lst      tab11.ixf  tab13.msg  tab16a.001.lob  tab17.msg  tab1.ixf
tab21.ixf  tab23.msg      tab25.msg  tab4a.001.xml  tab6.ixf  tab8.msg
EXPORT.out    tab11.msg  tab14.ixf  tab16.ixf      tab18.ixf  tab1.msg
tab21.msg  tab24.ixf      tab2.ixf   tab4.ixf      tab6.msg  tab9a.001.lob
tab10a.001.lob  tab12.ixf  tab14.msg  tab16.msg      tab18.msg  tab20.ixf
tab22.ixf  tab24.msg      tab2.msg   tab4.msg      tab7.ixf  tab9.ixf
tab10.ixf   tab12.msg  tab15.ixf  tab17a.001.xml  tab19.ixf  tab20.msg
tab22.msg  tab25a.001.xml  tab3.ixf   tab5.ixf      tab7.msg  tab9.msg
tab10.msg   tab13.ixf  tab15.msg  tab17.ixf      tab19.msg  tab21a.001.xml
tab23.ixf  tab25.ixf      tab3.msg   tab5.msg      tab8.ixf
```

After all of the required Db2 data is exported, copy the results from the db2move and db2look utilities from the source x86 system to the target Linux on IBM Z server. We used the rsync tool to complete this task.

Connected to our target lnx-z-db2 system, first create the database and its objects by using the DDL file that was generated by the db2look utility, as shown in Example 8-8.

Example 8-8 Creation of Db2 database objects

```
lnx-z-db2 $ db2 -tvf /db2_temp/SAMPLE.sql -z $HOME/SAMPLE_creation.log
```

Note: Review the output of the command that is shown in Example 8-8. Minor adjustments to the generated DDL file might be necessary, especially when migrating between different Db2 releases. In that case, DROP the database and retry the operation after reviewing the SQL statements. Refer to the Db2 product documentation for more information.

After the Db2 objects are migrated, load the data into the database by using the db2move utility. Finally, SET INTEGRITY on any tables that might require it, as shown in Example 8-9.

Example 8-9 Data restore using the db2move utility

```
lnx-z-db2 $ db2move SAMPLE load
lnx-z-db2 $ db2 connect to SAMPLE
lnx-z-db2 $ db2 "SELECT tabname,status,const_checked FROM syscat.tables WHERE
status='C'"
TABNAME STATUS CONST_CHECKED
-----
EMPLOYEE C      YYYYNYYYYYYYYYYYYYYYYYYYYYYYYYYY
DEPARTMENT C YYYYNYYYYYYYYYYYYYYYYYYYYYYYYYYY
2 record(s) selected.

lnx-z-db2 $ db2 SET INTEGRITY FOR EMPLOYEE,DEPARTMENT IMMEDIATE CHECKED
DB20000I The SQL command completed successfully.
```

We can now connect to the target database and query its data.

8.3 Migrating WebSphere Application Server Network Deployment cluster

IBM WebSphere Application Server Network Deployment is a full-featured middleware that supports various topologies and configurations and can integrate several aspects of your enterprise organization in accordance to your business needs.

A common migration strategy is to install and configure manually (or with the help of automated wsadmin's JACL or Jython scripts) all aspects of a cluster topology. In this scenario, the WebSphere Application Server cluster is re-created from scratch and all required components, such as JDBC data sources and JMS resources, are reconfigured. Finally, the application that uses these resources is redeployed in the new cluster and tested. This approach is typically done for simple applications that do not require much effort to be redeployed.

Another migration strategy involves the migration of the entire cell configuration directly from the source x86 servers to the new Linux on IBM Z systems. After the cell configuration is imported, the application is ready for use with minimal manual adjustments necessary. In this section, we migrate our WebSphere Application Server Network Deployment cell configuration by using this migration technique.

Our source WebSphere Application Server cell is composed of two servers in a horizontal topology. The deployment manager node is in xrhbres1, and this server also holds one node part of our application's cluster. The second node member of our cluster is in xrhbres2.

The first step is to back up our current profiles configuration, including the deployment manager and application servers, as shown in Example 8-10.

Example 8-10 Backup all WebSphere Application Server profiles before proceeding

```
xrhbres1 $ <DMGR_ROOT>/bin/backupConfig.sh /tmp/DmgrBackupBefore.zip -nostop
ADMU0116I: Tool information is being logged in file
           /opt/WebSphere/AppServer/profiles/Dmgr/logs/backupConfig.log
ADMU0128I: Starting tool with the Dmgr profile
ADMU5001I: Backing up config directory
           /opt/WebSphere/AppServer/profiles/Dmgr/config to file
           /tmp/DmgrBackupBefore.zip
```

```
(...) output suppressed (...)
ADMU5002I: 1,700 files successfully backed up

xrhrbres1 $ <APP_ROOT>/bin/backupConfig.sh /tmp/AppSrvrBackupBefore.zip -nostop
(...) output suppressed (...)

xrhrbres2 $ <APP_ROOT>/bin/backupConfig.sh /tmp/AppSrvrBackupBefore.zip -nostop
(...) output suppressed (...)
```

Because we are migrating our profiles to a different target architecture, we must ensure that the WebSphere Application Server release is under the same level at the source and target systems. Otherwise, we first must install the same level that we are migrating to into the source x86 system. Because we are performing an “as-is” migration, the WebSphere Application Server product is installed into our target Linux on IBM Z servers under the same level as our source x86 systems.

Note: For migrating between different WebSphere Application Server releases, see the WebSphere Application Server Network Deployment traditional documentation that is available at IBM Knowledge Center.

The next step is to transfer the profile’s configuration from our source x86 systems to our target Linux on IBM Z server. Because we are keeping the same topology, we transferred the deployment manager profile and the application server profile from `xrhrbres1` to `lnx-z-was-1`. We also copied the application server profile from `xrhrbres2` to `lnx-z-was-2`.

The next step is to create the profiles that compose our cluster topology into our Linux on IBM Z servers. It is important to keep the same node and cell names as we had on the source x86 systems. Example 8-11 shows the Deployment Manager profile creation at `lnx-z-was-1`.

Example 8-11 Create the target Deployment manager profile using the same cell and node names

```
lnx-z-was-1 $ <WAS_ROOT>/bin/manageprofiles.sh -create -profileName Dmgr
-profilePath /opt/WebSphere/AppServer/profiles/Dmgr -templatePath
/opt/WebSphere/AppServer/profileTemplates/management -serverType
DEPLOYMENT_MANAGER -nodeName AppCellNode -cellName AppCell -hostName lnx-z-was-1
-isDefault=false -enableAdminSecurity=false -disableWASDesktopIntegration

INSTCONFSUCCESS: Success: Profile Dmgr now exists. Please consult
/opt/WebSphere/AppServer/profiles/Dmgr/logs/AboutThisProfile.txt for more
information about this profile.
```

After the deployment manager profile is created, restore its configuration, as shown in Example 8-12.

Example 8-12 Deployment manager restore configuration

```
lnx-z-was-1 $ <DMGR_PROFILE_ROOT>/bin/restoreConfig.sh /tmp/DmgrBackupBefore.zip
ADMU0116I: Tool information is being logged in file
/opt/WebSphere/AppServer/profiles/Dmgr/logs/restoreConfig.log
ADMU0128I: Starting tool with the Dmgr profile
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: dmgr
ADMU2010I: Stopping all server processes for node AppCellNode
ADMU0512I: Server dmgr cannot be reached. It appears to be stopped.
ADMU5502I: The directory /opt/WebSphere/AppServer/profiles/Dmgr/config already
```

```
exists; renaming to
/opt/WebSphere/AppServer/profiles/Dmgr/config.old
ADMU5504I: Restore location successfully renamed
ADMU5505I: Restoring file /tmp/DmgrBackupBefore.zip to location
/opt/WebSphere/AppServer/profiles/Dmgr/config
ADMU5506I: 1,700 files successfully restored
ADMU6001I: Begin App Preparation -
ADMU6009I: Processing complete.
ADMU6002I: Begin Asset Preparation -
ADMU6009I: Processing complete.
```

The restored configuration still contains old references from the x86 servers. It is important that we update the internal deployment manager files to point to our new IBM Z servers. Example 8-13 shows how we updated the `serverindex.xml` file to point to our IBM Z topology. Three files must be updated: two for `lnx-z-was-1` and one for `lnx-z-was-2`.

Example 8-13 Update to `serverindex.xml` to reflect the new system names

```
lnx-z-was-1 $ find <DMGR_PROFILE_ROOT>/config/cells/ -name serverindex.xml
<DMGR_PROFILE_ROOT>/config/cells/AppCell/nodes/AppCellNode/serverindex.xml
<DMGR_PROFILE_ROOT>/config/cells/AppCell/nodes/AppNode/serverindex.xml
<DMGR_PROFILE_ROOT>/config/cells/AppCell/nodes/AppNode2/serverindex.xml
lnx-z-was-1 $ sed -i 's/xrhrbres1/lnx-z-was-1/g'
<DMGR_PROFILE_ROOT>/config/cells/AppCell/nodes/AppCellNode/serverindex.xml

lnx-z-was-1 $ sed -i 's/xrhrbres1/lnx-z-was-1/g'
<DMGR_PROFILE_ROOT>/config/cells/AppCell/nodes/AppNode/serverindex.xml

lnx-z-was-2 $ sed -i 's/xrhrbres2/lnx-z-was-2/g'
<DMGR_PROFILE_ROOT>/config/cells/AppCell/nodes/AppNode2/serverindex.xml
```

The final modification that we must perform is to update the x86 architecture references to the one that Linux on IBM Z uses. Example 8-14 provides a one-liner that does the job for us, and checks the correct values to use.

Example 8-14 Update `node-metadata.properties` architecture

```
lnx-z-was-1 $ find <DMGR_ROOT>/config/cells/ -name node-metadata.properties -exec
grep x86_64 {} \;
com.ibm.websphere.sdk.architecture.8.0_64=x86_64
com.ibm.websphere.sdk.nativeLibPath.8.0_64=${WAS_INSTALL_ROOT}/lib/native/linux/x8
6_64/
com.ibm.websphere.sdk.nativeLibPath.8.0_64=${WAS_INSTALL_ROOT}/lib/native/linux/x8
6_64/
com.ibm.websphere.sdk.architecture.8.0_64=x86_64
com.ibm.websphere.sdk.nativeLibPath.8.0_64=${WAS_INSTALL_ROOT}/lib/native/linux/x8
6_64/
com.ibm.websphere.sdk.architecture.8.0_64=x86_64

lnx-z-was-1 $ find <DMGR_ROOT>/config/cells/ -name node-metadata.properties -exec
sed -i 's/x86_64/s390_64/g' {} \;
```

Note: From this point on, it is recommended that you shut down your x86 cluster to avoid problems. Do not proceed if you failed to perform any of the previous steps.

It is now time to start the deployment manager for the cell, as shown in Example 8-15.

Example 8-15 Deployment Manager start

```
lnx-z-was-1 $ <DMGR_ROOT>/bin/startManager.sh
ADMU0116I: Tool information is being logged in file
           /opt/WebSphere/AppServer/profiles/Dmgr/logs/dmgr/startServer.log
ADMU0128I: Starting tool with the Dmgr profile
ADMU3100I: Reading configuration for server: dmgr
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server dmgr open for e-business; process id is 294578
```

After the deployment manager process is started, create the application server profiles, restore their respective configurations, and synchronize them with the new deployment manager. Example 8-16 shows how this process was done under lnx-z-was-2. Remember to use the same cell and node names as the source x86 system for the profiles that you create.

Example 8-16 Application server configuration restore and synchronization with the new cell

```
lnx-z-was-2 $ <WAS_INSTALL_ROOT>/bin/manageprofiles.sh -create -profileName
AppSrvr -profilePath /opt/WebSphere/AppServer/profiles/AppSrvr -templatePath
/opt/WebSphere/AppServer/profileTemplates/managed -nodeName AppNode2 -cellName
AppCell1 -hostName lnx-z-was-2 -enableAdminSecurity=false -federateLater=true
-disableWASDesktopIntegration
INSTCONFSUCCESS: Success: Profile AppSrvr now exists. Please consult
/opt/WebSphere/AppServer/profiles/AppSrvr/logs/AboutThisProfile.txt for more
information about this profile.

lnx-z-was-2 $ <PROFILE_ROOT>/bin/restoreConfig.sh /tmp/AppSrvrBackupBefore.zip
ADMU0116I: Tool information is being logged in file
           /opt/WebSphere/AppServer/profiles/AppSrvr/logs/restoreConfig.log
ADMU0128I: Starting tool with the AppSrvr profile
ADMU0507I: No servers found in configuration under:

/opt/WebSphere/AppServer/profiles/AppSrvr/config/cells/AppCell1/nodes/AppNode2/serv
ers
ADMU2010I: Stopping all server processes for node AppNode2
ADMU5502I: The directory /opt/WebSphere/AppServer/profiles/AppSrvr/config
           exists; renaming to
           /opt/WebSphere/AppServer/profiles/AppSrvr/config.old
ADMU5504I: Restore location successfully renamed
ADMU5505I: Restoring file /tmp/AppSrvrBackupBefore.zip to location
           /opt/WebSphere/AppServer/profiles/AppSrvr/config
ADMU5506I: 2,137 files successfully restored
ADMU6001I: Begin App Preparation -
ADMU6009I: Processing complete.
ADMU6002I: Begin Asset Preparation -
ADMU6009I: Processing complete.

lnx-z-was-2 $ <PROFILE_ROOT>/bin/syncNode.sh lnx-z-was-1 8879
ADMU0116I: Tool information is being logged in file
           /opt/WebSphere/AppServer/profiles/AppSrvr/logs/syncNode.log
ADMU0128I: Starting tool with the AppSrvr profile
(...) output suppressed (...)
ADMU0401I: Begin syncNode operation for node AppNode2 with Deployment Manager
lnx-z-was-1: 8879
```

```

ADMU0016I: Synchronizing configuration between node and cell.
ADMU0402I: The configuration for node AppNode2 has been synchronized with
           Deployment Manager Inx-z-was-1: 8976

```

You can now start your application server nodes, as shown in Example 8-17.

Example 8-17 WebSphere Application Server node start-up

```

Inx-z-was-1 $ <APP_PROFILE_ROOT>/bin/startNode.sh
ADMU0116I: Tool information is being logged in file
/opt/WebSphere/AppServer/profiles/AppSrvr/logs/nodeagent/startServer.log
ADMU0128I: Starting tool with the AppSrvr profile
ADMU3100I: Reading configuration for server: nodeagent
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server nodeagent open for e-business; process id is 289634

Inx-z-was-2 $ <APP_PROFILE_ROOT>/bin/startNode.sh
(...) output suppressed (...)

```

The migration of WebSphere Application Server is now complete. You now can access the administrative console and check the status of your cluster. The user name and password credentials to authenticate typically are the same as in the source x86 system. Figure 8-1 shows our WebSphere Application Server topology that is fully synchronized with our target Linux on IBM Z servers.

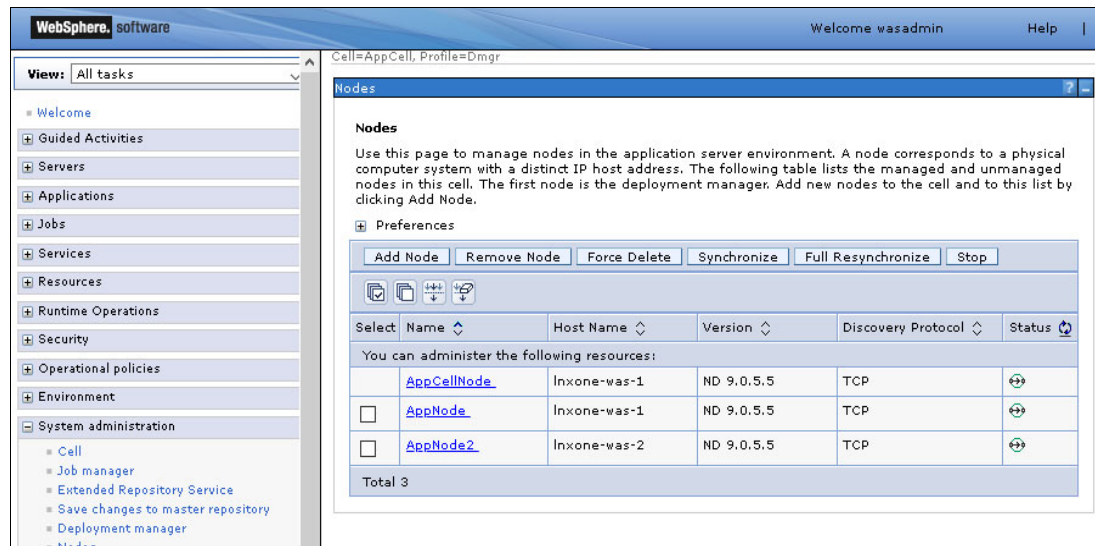


Figure 8-1 WebSphere migrated cell into Linux on IBM Z

Note: For enhanced security, it is recommended to modify the credentials to the administrative consoles and to update the profile's property files, such as the `soap.client.props` and `ssl.client.props` files. For more information, see your product documentation.

Before starting our application, we updated our data sources to reflect the new Db2 location. Then, we tested the connection to ensure that everything was correctly set up.

8.4 Migrating WebSphere Application Server Liberty and Open Liberty

[WebSphere Liberty](#) stands as a leading runtime solution catering to the creation of new cloud-native applications and the modernization of existing workloads. Its lightweight and efficient design make it particularly suitable for microservices, while its extensive API support and adaptable architecture position it as an optimal choice for modernization efforts. WebSphere Liberty, including its open-source version Open Liberty®, caters to modern development and operational needs, serving as an excellent option for updating applications from traditional servers like WebSphere Application Server.

Developers benefit from a streamlined experience that enhances productivity, offering integrated development and testing capabilities both with and without containers. Its design aligns seamlessly with modern automation practices like continuous integration and continuous deployment (CI/CD).

Operations teams, DevOps practitioners, can use the cost-effectiveness of deploying both microservices and monolithic applications with WebSphere Liberty. Its simplified deployment process requires no tuning, and it seamlessly integrates with Kubernetes for deployment and management.

[Open Liberty](#) (an IBM Open Source Project) adheres to the Eclipse Public License (EPL) and supports Java Enterprise Edition (EE)/Jakarta EE Full Platform 7 and 8, as well as MicroProfile 1 to 4.1.

Migrating WebSphere Application Server Liberty and Open Liberty:

While traditional migration approaches exist, embracing CI/CD (Continuous Integration and Continuous Delivery) patterns unlocks a more agile and efficient way to move your WebSphere Liberty applications from x86 to Linux on IBM Z. This section explores two potential migration paths:

- ▶ WebSphere Application Server Liberty installation and pipeline deployment:
 - On Linux on IBM Z, install the WebSphere Liberty runtime.
 - Integrate your application and artifacts into your existing CI/CD pipeline. The pipeline then handles the deployment process on the target Linux on IBM Z environment, streamlining automation and consistency.
- ▶ WebSphere Application Server Liberty installation and manual migration:
 - On Linux on IBM Z, install the WebSphere Liberty runtime.
 - From the source (x86) environment, copy the application directory and the `server.xml` configuration file to the Linux on IBM Z server.
 - Modify the `server.xml` file to reflect the new Linux on IBM Z server's host name or IP address.

This section will showcase the second alternative for illustrative purposes, demonstrating a simplified migration approach leveraging pre-configured components. However, it's crucial to carefully evaluate both options and choose the one that best aligns with your specific application needs and existing infrastructure.

Regardless of the chosen path, CI/CD integration plays a vital role in streamlining the migration process, ensuring consistency, and minimizing manual intervention. This approach paves the way for faster deployments, reduced errors, and enhanced agility in your migration journey.

To perform a manual migration of WebSphere Application Server Liberty / Open Liberty follow those steps:

- Download and install the official release of [IBM Semeru Runtime Open Edition for Java 21](https://github.com/ibmruntimes/semeru21-binaries/releases/download/jdk-21.0.2%2B13_openj9-0.43.0/ibm-semeru-open-21-jdk-21.0.2.13_0.43.0-1.s390x.rpm) onto the Linux on IBM Z LPAR as shown in Example 8-18. It's important to ensure that you're using the same version as the source server (x86).

Example 8-18 Install IBM Semeru Runtime Open Edition for Java

```
[root@lnx-z-1 ~]# rpm -ivh
https://github.com/ibmruntimes/semeru21-binaries/releases/download/jdk-21.0.2%2B13
_openj9-0.43.0/ibm-semeru-open-21-jdk-21.0.2.13_0.43.0-1.s390x.rpm
```

- Set IBM Semeru Runtime as the default Java runtime environment on Linux on IBM Z as shown in Example 8-19.

Example 8-19 Setting default Java runtime

```
[root@lnx-z-1 ~]# update-alternatives --config java
```

There are 2 programs which provide 'java'.

Selection	Command

*+ 1	java-11-openjdk.s390x (/usr/lib/jvm/java-11-openjdk-11.0.22.0.7-2.el9.s390x/bin/java)
2	/usr/lib/jvm/ibm-semeru-open-21-jdk/bin/java

Enter to keep the current selection[+], or type selection number: 2

- Verify the Java version as shown in Example 8-20.

Example 8-20 Verify Java version

```
[root@lnx-z-1 ~]# java -version
openjdk version "21.0.2" 2024-01-16 LTS
IBM Semeru Runtime Open Edition 21.0.2.0 (build 21.0.2+13-LTS)
Eclipse OpenJ9 VM 21.0.2.0 (build openj9-0.43.0, JRE 21 Linux s390x-64-Bit
Compressed References 20240116_94 (JIT enabled, AOT enabled)
OpenJ9      - 2c3d78b48
OMR         - ea8124dbc
JCL         - 78c4500a434 based on jdk-21.0.2+13)
[root@lnx-z-1 ~]#
```

- Download Open Liberty 24.0.0.2 as shown in Example 8-21. Ensure it matches the version used on the source server.

Example 8-21 Download Open Liberty packages

```
[root@lnx-z-1 ~]# wget
https://public.dhe.ibm.com/ibmdl/export/pub/software/openliberty/runtime/release/2
4.0.0.2/openliberty-jakartaee10-24.0.0.2.zip
--2024-03-22 14:35:10--
https://public.dhe.ibm.com/ibmdl/export/pub/software/openliberty/runtime/release/2
4.0.0.2/openliberty-jakartaee10-24.0.0.2.zip
Resolving public.dhe.ibm.com (public.dhe.ibm.com)... 170.225.126.18
Connecting to public.dhe.ibm.com (public.dhe.ibm.com)|170.225.126.18|:443...
connected.
```

```
HTTP request sent, awaiting response... 200 OK
Length: 133034245 (127M) [application/zip]
Saving to: 'openliberty-jakartaee10-24.0.0.2.zip'
```

```
openliberty-jakartaee10-24.0.0.2.zip.2
100%[=====] 126.87M
4.58MB/s   in 19s
```

```
2024-22-03 14:35:31 (6.82 MB/s) - 'openliberty-jakartaee10-24.0.0.2.zip' saved
[133034245/133034245]
```

-
- Uncompress the installation package archive as shown in Example 8-22.

Example 8-22 Unzip installation archive

```
[root@lnx-z-1 tmp]# unzip openliberty-jakartaee10-24.0.0.2.zip
```

- With the necessary components installed and configured, create and start a new server Open Liberty instance on the Linux on IBM Z environment as shown in Example 8-23.

Example 8-23 Create and start Open Liberty instance

```
[root@lnx-z-1 bin]# /root/wlp/bin/server create redbookserver1
```

```
Server redbookserver1 created.
```

```
[root@lnx-z-1 bin]# /root/wlp/bin/server start redbookserver1
```

```
Starting server redbookserver1.
```

- To illustrate the migration process, we have created a simple application that displays a message and shows the server CPU architecture as shown in Example 8-24.

Example 8-24 Simple web application

```
@Path("properties")
```

```
public class HelloWorld {
```

```
    @GET
```

```
    @Produces(MediaType.APPLICATION_JSON)
```

```
    public String getProperties() {
```

```
        String message = "Hello IBM Residency Team - ZS-EP02 Practical migration
from x86 to Linux on IBM Z. The application is currently running from : "
```

```
        + System.getProperty("os.arch");
```

```
        return message;
```

```
    }
```

```
}
```

- Place the application in the **apps** directory and modify the **server.xml** file to reflect the application's path and context as shown in Example 8-25 on page 240.

Example 8-25 server.xml on the source server (x86)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<server description="new server">
```

```
    <!-- Enable features -->
```

```
    <featureManager>
```

```
        <feature>jakartaee-10.0</feature>
```



```

        <feature>microProfile-6.1</feature>
        <feature>restfulWS-3.1</feature>
        <feature>jsonb-3.0</feature>
    </featureManager>

    <!-- This template enables security. To get the full use of all the
    capabilities, a keystore and user registry are required. -->

    <!-- For the keystore, default keys are generated and stored in a keystore. To
    provide the keystore password, generate an
        encoded password using bin/securityUtility encode and add it below in the
    password attribute of the keyStore element.
        Then uncomment the keyStore element. -->
    <!--
    <keyStore password="" />
    -->

    <!--For a user registry configuration, configure your user registry. For
    example, configure a basic user registry using the
        basicRegistry element. Specify your own user name below in the name
    attribute of the user element. For the password,
        generate an encoded password using bin/securityUtility encode and add it
    in the password attribute of the user element.
        Then uncomment the user element. -->
    <basicRegistry id="basic" realm="BasicRealm">
        <!--
        <user name="yourUserName" password="" />
        -->
    </basicRegistry>

    <!-- To access this server from a remote client add a host attribute to the
    following element, for example host="*" -->
    <httpEndpoint id="defaultHttpEndpoint"
        host="yslserver"
        httpPort="9080"
        httpsPort="9443" />

    <!-- Automatically expand WAR files and EAR files -->
    <applicationManager autoExpand="true"/>

    <!-- Configures the application on a specified context root -->
    <webApplication contextRoot="{app.context.root}" location="demoapp.war" />

    <!-- Default SSL configuration enables trust for default certificates from the
    Java runtime -->
    <ssl id="defaultSSLConfig" trustDefaultCerts="true" />

</server>
~

```

-
- To test the application on the source serve (x86), use the **curl** command as shown in Example 8-26.

Example 8-26 Test application from source server

```
[root@yslserver bin]# curl http://yslserver:9080/LibertyProject/system/properties
Hello IBM Residency Team - ZS-EP02 Practical migration from x86 to Linux on IBM Z.
The application is currently running from : aarch64
```

- Copy the **server.xml** and **apps** directory from the source server to Linux on IBM Z. Note that 'yslserver' represents the Open Liberty server on the source (x86), while 'lnx-z-1' refers to the Open Liberty server on the target (Linux on IBM Z) as shown in Example 8-27.

Example 8-27 Copy server.xml and apps directory from source to target

```
[root@yslserver bin]# scp -i ysl.pem /root/wlp/usr/servers/yslsource/server.xml
zlinux@lnx-z-1:/root/wlp/usr/servers/redbookserver1
```

```
[root@yslserver bin]# scp -i ysl.pem -r /root/wlp/usr/servers/yslsource/apps/
zlinux@lnx-z-1:/root/wlp/usr/servers/redbookserver1
```

- Modify the **server.xml** file and update the host name accordingly as shown in Example 8-28.

Example 8-28 server.xml on the target server (Linux on IBM Z)

```
[root@lnx-z-1 bin]# scp -i ysl.pem /root/wlp/usr/servers/yslsource/server.xml
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
```

```
    <!-- Enable features -->
    <featureManager>
        <feature>jakartaee-10.0</feature>
        <feature>microProfile-6.1</feature>
        <feature>restfulWS-3.1</feature>
        <feature>jsonb-3.0</feature>
    </featureManager>
```

```
    <!-- This template enables security. To get the full use of all the
capabilities, a keystore and user registry are required. -->
```

```
    <!-- For the keystore, default keys are generated and stored in a keystore. To
provide the keystore password, generate an
        encoded password using bin/securityUtility encode and add it below in the
password attribute of the keyStore element.
```

```
    Then uncomment the keyStore element. -->
    <!--
    <keyStore password=""/>
    -->
```

```
    <!--For a user registry configuration, configure your user registry. For
example, configure a basic user registry using the
        basicRegistry element. Specify your own user name below in the name
attribute of the user element. For the password,
        generate an encoded password using bin/securityUtility encode and add it
in the password attribute of the user element.
```

```
    Then uncomment the user element. -->
    <basicRegistry id="basic" realm="BasicRealm">
    <!--
```

```

        <user name="yourUserName" password="" />
    -->
</basicRegistry>

<!-- To access this server from a remote client add a host attribute to the
following element, for example host="*" -->
<httpEndpoint id="defaultHttpEndpoint"
    host="lnx-z-1"
    httpPort="9080"
    httpsPort="9443" />

<!-- Automatically expand WAR files and EAR files -->
<applicationManager autoExpand="true"/>

<!-- Configures the application on a specified context root -->
<webApplication contextRoot="${app.context.root}" location="demoapp.war" />

<!-- Default SSL configuration enables trust for default certificates from the
Java runtime -->
<ssl id="defaultSSLConfig" trustDefaultCerts="true" />

</server>

```

- Verify that the application is correctly installed and functioning as expected on the Linux on IBM Z platform as shown in Example 8-29.

Example 8-29 Test application from target server

```
[root@lnx-z-1 bin]# curl http://lnx-z-1:9080/LibertyProject/system/properties
Hello IBM Residency Team - ZS-EP02 Practical migration from x86 to Linux on IBM Z.
The application is currently running from : s390x
```

Note: You can also copy dropping directory from source to target. By default, the dropins directory is automatically monitored. If you drop an application into this directory, the application is automatically deployed on the server.

8.5 Migrating Fibre Channel devices

Disk storage on servers often is made available by using a storage area network (SAN). A widely spread technology to distribute storage devices is Fibre Channel.

In the distributed world, the term Fibre Channel is often abbreviated as FC. FC in mainframe terminology stands for FICON, so a new abbreviation was introduced, FCP, which stands for Fibre Channel Protocol.

Typically, a SAN with Fibre Channel consists of two independent fabrics, which are point-to-point connectivity between processor and peripheral devices. All the adapters have their own unique worldwide number (WWN) which is put into a zone within the fabric. The servers typically have two interfaces that reside in different fabrics.

Modern Fibre Channel adapters can be virtualized by using N_Port ID Virtualization (NPIV). They provide a number of different virtual devices that all have their unique WWN and thus can be put into their specific zone. Distributed servers that have only one operating system

commonly do not need this feature because the base WWN is sufficient to make all needed disk devices available.

When consolidating multiple servers to a single server, NPIV is a method to separate the disk devices for the different servers. These separations are important, and NPIV makes the separation possible even though devices are attached over the same physical connection. This holds true for the mainframe, where the FCP devices can be switched to NPIV mode. Unlike distributed systems, the mainframe has its own idea about which WWN is used for which device number. It is important to understand that the WWNs also differ from machine to machine, and when migrating from a distributed environment to a mainframe, an update of the zoning and of the storage system normally is needed. These changes are described in detail below.

8.5.1 Zoning for FCP

To connect the NPIV adapters to the respective logical unit number (LUN) on the storage device, the Fibre Channel switches must support zoning, and zoning must be set up accordingly. Each adapter must be added to a zone that also contains the adapter of the storage device. The storage then selects the correct disk by using the NPIV WWN.

In theory, just one zone with all adapters and storage adapters would be sufficient. For actual production deployments, create a separate zone for each of the NPIV devices. The reason is that during logon and logoff of a single NPIV device, the whole zone is rediscovered. While this does not cause errors, it still can cause short hangs depending on the size of the zone. If a separate zone is created for each NPIV device, only the local zone is discovered, which has no affect on other zones.

8.5.2 FCP and multipath

The failover configuration for FCP is not handled by PR/SM or z/VM, but must be done from within Linux for IBM Z. Therefore, two NPIV adapters must be attached to the guest system that is connected over the two different Fibre Channel fabrics.

The multipath setup itself is configured inside the Linux guest system with the configuration file `/etc/multipath.conf`. After the `multipathd` daemon is started, all available LUNs together with their paths can be checked with the command:

```
multipath -ll
```

Note regarding SLES: The behavior of SCSI devices changes between SLES11 and SLES12. In SLES11, all LUNs had to be configured entirely manually. With SLES12, automatic LUN scanning has been switched on, and therefore all LUNs will automatically be detected after the `zfc` host has been configured.

The actual configuration of multipath depends on the storage device used. For DS8000 storage systems, the configuration shown in Example 8-30 can be used.

Example 8-30 multipath.conf sample configuration file

```
defaults {  
    path_grouping_policy multibus  
    failback 1  
    rr_min_io 10  
    path_checker tur
```

```

    checker_timeout 60
}

devices {
    device {
        vendor "IBM"
        product ""
        path_grouping_policy group_by_prio
        prio alua
    }
}

```

Multipath is needed when using FCP to prevent disk failures. Without multipath, any failure in a host bus adapter (HBA), fiber optic cable, or transceiver can cause unrecoverable data loss. Multipath is needed for normal operations, when important updates to the HBA microcode are required. Disk traffic can continue on one path while the other path is down for the update. Without multipath, all services involving the FCP disk would have to be halted while upgrading the microcode.

8.5.3 FCP migration setup tasks

The migration of Fibre Channel devices from a distributed system to the mainframe with NPIV involves several different tasks:

1. Dedicate NPIV adapters

Assuming that NPIV adapters already have been configured within the CEC, a pair of NPIV adapters that is attached to the two different zones must be dedicated to the new guest system. For redundancy reasons, those adapters should come from two different physical cards.

It is good practice to always dedicate the same virtual device (vdev) number. For example, if you configured two device ranges FA00-FA1D and FC00-FC1D for the two fabrics, dedicate the same generic vdevs as pairs for each guest:

DEDICATE FA00 FA06

DEDICATE FC00 FC06

That way, all the virtual addresses are always FA00 and FC00 and the numbering is relatively obvious.

2. NPIV WWNs

To retrieve the WWNs of the respective NPIV adapter, proceed as follows:

- a. Note the name of the LPAR that runs your z/VM system.
- b. Log on to the IBM Z Support Element (SE) of the mainframe, either via the Hardware Management Console (HMC) with “Single Object Operations”, or directly.
- c. On the SE, find the NPIV adapter information at “System Management” → <system name> → CPC Configuration → FCP Configuration. Where <system name> is your system name.
- d. The easiest way to retrieve the information is to transfer the file with the WWNs for the z/VM LPAR to a remote FTP server via FTP.
- e. The resulting file is a comma-separated list, which looks like the following:

```
LP1,00,01,14,00,f91c,c05076e0f3002d70,0n,Yes,05a0,c05076e0f3005a01
```

- f. To retrieve the WWNs for “fa06” and “fc06” on LP1, use the following command:

```
# grep LP1 -list.csv | grep -e fa06 -e fc06 | cut -d, -f 6,7
fa06,c05076e0f3000798
fc06,c05076e0f3001e18
```

3. Zoning update

The WWNs that have been retrieved from the SE must be used to update the Fibre Channel zones. If all servers already have their own individual zone, just add the WWNs from step 2 to the respective fabric. If just one big zone exists at this time, split the zone up into smaller zones for each server during the process. The intended result is to have a pair of zones for the two fabrics for each of the migrated servers.

4. Storage system update

Inside the storage system, the host connections are configured according to the WWN of the Fibre Channel adapter. Without NPIV, this is just the base number. While with NPIV, the respective NPIV WWN is used. Normally, it is possible to add several host connections to a specific storage group. This allows several Fibre Channel adapters to access the same LUNs. To prepare a migration, add the new WWN to the configured host connections. After the restart of the service on the mainframe has finished, remove the old WWN from the host connection.



Postmigration considerations

This chapter describes general postmigration consideration concepts for getting acceptance, measuring performance, and tuning. Topics that are covered in this chapter include an acceptance list, performance measurement understanding, and key considerations for performance tuning.

Every migration poses a large challenge for IT organizations because each stakeholder has different expectations and requirements from the project. Most of the topics after migration center around performance and functionality. IT organizations face the following difficult questions:

- ▶ What exactly has been done?
- ▶ Is there anything missing?
- ▶ Is everything working?
- ▶ Is the performance as expected?
- ▶ Is the process completed?
- ▶ Did we get approvals?

To answer these questions, take steps before and after the migration implementation phase.

This chapter includes the following sections:

- ▶ 9.1, “Gaining acceptance” on page 248
- ▶ 9.2, “Performance measurement” on page 248
- ▶ 9.3, “Performance tuning” on page 250

9.1 Gaining acceptance

Migration projects are generally recognized as major changes to the IT environment. Each change requires significant test and acceptance by various stakeholders. Decisions must be made by these stakeholders whether the migration was a success.

Acceptance requires an understanding of the big picture, before and after migration:

- ▶ Before implementation phase start:
 - Decide and document test scope.
 - Decide and document test case (including test scenario).
 - Create postmigration checklist for all components.
 - Collect performance data on system.
 - Get acceptance from stakeholder for test.
- ▶ After implementation done:
 - Use post-migration checklist and check whether implementation done or not.
 - Test system using by documented test case. Complete and document all scenarios.
 - Measure performance and compare with previous performance data.
 - If necessary, perform performance tuning.

Based on project scope and context, items used for acceptance testing can change but the following list is the most common acceptance tests performed before gaining stakeholder acceptance:

- ▶ Application testing (In some cases usability testing may be required.)
- ▶ Functional testing
- ▶ Performance testing
- ▶ Security testing
- ▶ User acceptance testing

9.2 Performance measurement

In this section, we describe performance measurement and its impact on the success of your migration. The most important point to consider is that you need to measure the performance of the application when it is running in production on the source environment and then compare that with the performance of the application on the target environment.

We also describe monitoring commands and tools that can assist you in identifying and resolving performance inhibitors.

9.2.1 What is performance

“Performance” in computer systems is very much a relative term. Usually computer performance is described in measurable terms, such as transactions per second, response time, time to process a booking or insurance sale. However, when a migration project is undertaken, it is important to understand the performance metrics used on the source environment so that you can understand the relative performance on the target system.

The initial performance of a new system may often not be as expected especially when changing hardware platforms. Therefore, tuning must be undertaken to improve the performance of the target system. Without having proper metrics, it is impossible to validate the performance of the new platform relative to the former platform.

For this reason, the migration project team first needs to agree on what performance metrics from the source platform will be used in the migration project plan to measure the performance of the target platform.

9.2.2 Choosing what to measure

To determine the success of a migration, simply having the application on the target platform provide the same answers as the source platform does not prove success. The natural expectation of a migration onto Linux on IBM Z is that the application will not only be more resilient and available because of IBM Z, but that it will also provide equal or better performance than the source platform. To ensure that the performance improvements are easy to show, it is important to choose the right metrics. But what are these metrics, and how should they be measured?

Response time

Response time is the measure of the time it takes for something to happen in a computer system. Generally we choose to measure the response time of a unit of work called a unit-of-work ID. This could entail something as simple as checking an account balance, to something as complex as the time taken to issue a new insurance policy or open a new bank account.

The point to remember with computer systems is that the response time of a single transaction is the sum of a number of response times. Figure 9-1 shows the various components that make up user response time.

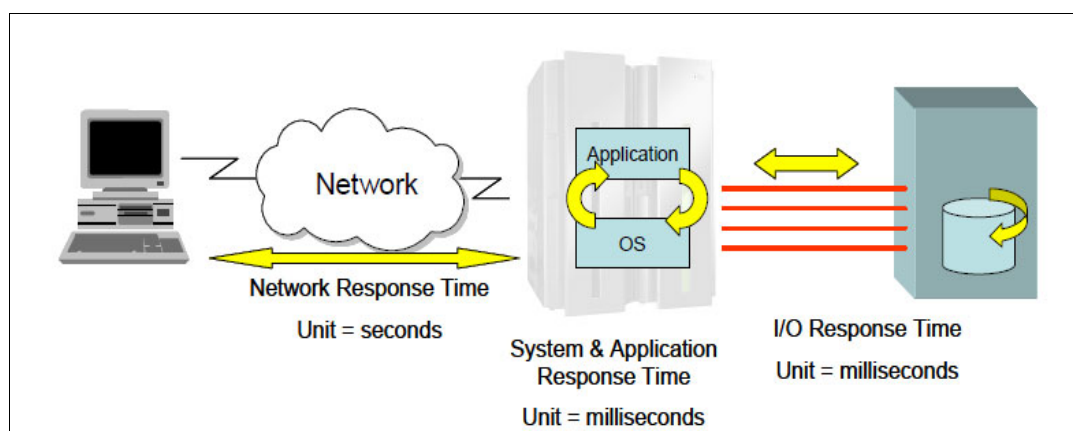


Figure 9-1 Components that make up the response time of transactions

The figure shows that there are two points where response time could be measured: system response time and user response time. When you are trying to understand the relative performance improvement from a new system, the only point to measure response time is from when a system receives the request and when it provides a response of some sort to the request.

In the case illustrated by Figure 9-1, the system response time will include application time and the I/O response time to access the data. If you choose to measure the response time of user experiences at their terminal or over the web, you will be adding in the network response time, which can vary greatly for the same transaction because it can be influenced by network load.

To compare the source and target systems directly, the recommended approach is to measure system response time on the source system, and assuming the application has not changed greatly, measure the system response time on the target platform.

Transaction throughput

The transaction throughput performance metric may provide a more meaningful measure of system performance because it measures the number of transactions processed over a period of time. This is typically one second, but could be any time period that you prefer.

In both cases, you should have baseline performance metrics for the source system to properly compare both the new and old systems.

9.3 Performance tuning

Tuning any system should follow some principles because every hardware and software platform has unique features and characteristics that must be considered when you tune your environment. The art of tuning performance in a system to require success performance analyses, multi-step tuning process and change management strict combination.

Regardless of which tools you choose, the best methodology for analyzing the performance of a system is to start from the outside and work way down to the small tuning details in the system. Start gathering data about overall health of the system hardware and processes. The following list is a sampling of the types of questions you should answer about both your source and target systems:

- ▶ How busy is the processor during the peak periods of each day?
- ▶ What happens to I/O response times during those peaks?
- ▶ Do they remain fairly consistent, or do they elongate?
- ▶ Does the system get memory constrained every day, causing page waits?
- ▶ Can current system resources provide user response times that meet service level agreements?

It is important to know what tuning tools are available and what type of information they provide. Equally important is knowing when to use those tools and what to look for. How will you know what is normal for your environment and what is problematic unless you check the system activity and resource utilization regularly? Conducting regular health checks on a system also provides utilization and performance information that you can use for capacity planning.

Tuning is not a one-size-fits-all approach, as a system tuned for one type of workload performs poorly with another type of workload. This means that you must understand the workload that you want to run and be prepared to review your tuning efforts when the workload changes. A simple workload is a server that shows one or more peaks during the day, while a complicated workload is an application that is CPU-intensive during part of the day and I/O-intensive during another part. The most cost-efficient approach to running these workloads is to adjust the capacity of the server during the day. This is exactly what z/VM carries out. Portions of the virtual machine are brought in to run in main memory while inactive virtual machines are moved to paging to create space.

Multi-step tuning process requires the skills of a systems performance detective. A systems performance analyst identifies IT problems using a detection process similar to that of solving a crime. In IT systems performance, the crime is a performance bottleneck or sudden degrading response time.

The performance analyst asks questions, searches for clues, researches sources and documents, reaches a hypothesis, tests the hypothesis by tuning or other means, and eventually solves the mystery, which results in improved system performance. Bottleneck analysis and problem determination are facilitated by sophisticated tools such as:

- ▶ IBM Tivoli OMEGAMON® on z/VM and Linux: OMEGAMON detects performance problems and alerts you before degraded response time becomes evident. OMEGAMON detects a potential performance or availability problem and sends a warning alert for z/VM and OMEGAMON console.
- ▶ IBM Instana platform: Instana help to identify and isolate issues and decrease MTTR for mission critical applications and provides a complete end-to-end view, enabling application owners and site reliability engineers (SREs) the ability to detect and isolate potential problems from any part of the application:
 - Understand the health of hybrid multi-cloud applications in a single view
 - Integrate tracing and metric data from both agents and Open Telemetry providing the best of both worlds and extending from end-to-end, including IBM Z
 - Empower application teams to detect and isolate performance issues even with limited knowledge

For more information, see [Getting started with Instana](#).

Figure 9-2 provides an overview of Instana capabilities for Linux on IBM Z / LinuxONE.

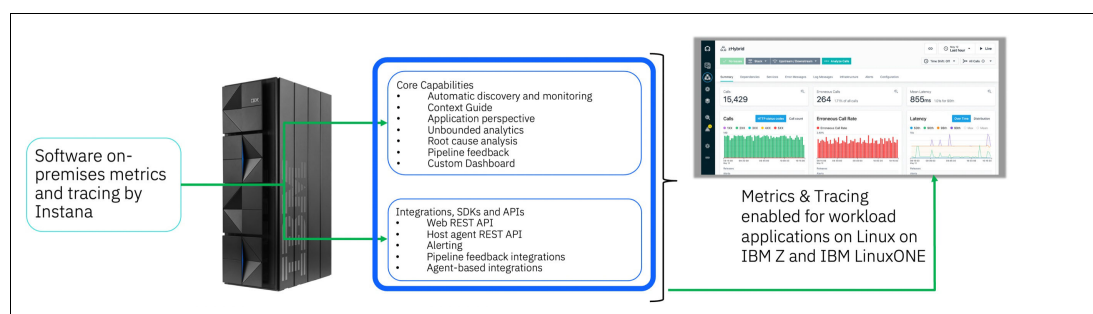


Figure 9-2 Instana capabilities for Linux on IBM Z / LinuxONE

Change management that is not strictly related to performance tuning is probably the single most important factor for successful performance tuning. The following considerations highlight this point:

- ▶ Implement a proper change management process before tuning any system.
- ▶ Never start tweaking settings on a production system.
- ▶ Never change more than one variable at a time during the tuning process.
- ▶ Retest parameters that supposedly improve performance; sometimes statistics come into play.

Document successful parameters and share them with the community no matter how trivial you think they are. System performance can benefit greatly from any results obtained in various production environments.



Additional use case scenarios

The complexity of a migration from Linux on the x86 may change by platform architecture and context of the migration. The Linux operating system is more straightforward and well-known and makes the possibility for migration much easier for technical people. However, when you consider an application, database management system, or middleware migration, you need to consider degrees of complexity, cost, and risk.

In this appendix, we describe additional use case scenarios where a telecommunications company, a healthcare company, and a financial data provider company all want to migrate from x86 to Linux on IBM Z. We discuss the challenges inherent to each industry and describe their respective migration scenarios.

Telecommunication Company using Linux on IBM Z Crypto Capabilities for Cybersecurity

In this scenario, the fictional telecommunication company, Fictional Telco Company T1, selects the IBM Z platform as their platform for running critical and secure workloads. Security is a key concern in a scenario where personal and sensitive data are protected by government regulations and have become the main asset of most companies. Telco T1 wants to protect all the data by pervasively encrypting data at rest and data in flight and getting safe against new quantum attacks. In this example, the following technology can be employed:

Consolidated hardware infrastructure:

- ▶ IBM Z z16
- ▶ Crypto Express8 Cards
- ▶ The Trusted Key Entry (TKE) workstation
- ▶ IBM Z/VM 7.3
- ▶ Red Hat Enterprise Linux Servers on IBM Z platform

Building a secure platform for sensitive workloads:

- ▶ IBM Crypto Express adapters are tamper-responding Hardware Security Modules (HSMs) that support cryptographic operations using secure keys.
- ▶ The HSMs configured as Common Cryptographic Architecture (CCA) adapters are intended for the financial industry and are certified as payment card industry (PCI) compliant as well as Enterprise PKCS #11 (EP11) adapters are intended for workloads using the PKCS#11 standard.
- ▶ Each processor of a LinuxONE system has a special component called Central Processor Assist for Cryptographic Functions (CPACF) which accelerates the most common cryptographic operations that are standardized by the US National Institute of Standards and Technology (NIST), for example AES, SHA2, SHA3, ECDH, and ECDSA.
- ▶ The LinuxONE next model will use quantum safe methods inside its hardware and firmware to protect customer hardware investments against potential quantum threats, by providing first versions of quantum-safe cryptographic algorithms accessible to Linux software.
- ▶ The Trusted Key Entry (TKE) workstation is an optional feature that acts as an alternative to clear key entry. You can use the TKE workstation to load master keys, and operational keys in a secure and controlled way.

On the software side, all the Linux file systems were built with pervasive encryption by leveraging Linux Unified Key Setup (LUKS) and dm-crypt using protected-keys accelerated and protected by Crypto Express HSM.

Data in flight protection were achieved by enabling and configuring OpenSSL and TLS (Transport Layer Security) in all network communications.

Cryptography operations were also used inside the application for digitally signing documents, accelerating hashes and kernel entropy of random numbers.

By choosing IBM Z as a prime secure platform, telecom T1 achieved the highest levels of security compliance and offered a highly secure platform, as well as maintaining high performance and reliability. As the heavy processing of crypto operations is off-loaded to specialized hardware and firmware components, the general performance of the workload is not heavily penalized as it is in x86 environments.

Healthcare industry: Mobile and Internet solution

In this scenario, the fictional healthcare company, Fictional Hospital H1, also chooses Linux on IBM Z as its mobile application platform. Hospital H1 wants to build a secure platform, increase responsiveness, and value perception, and reduce multi-platform development costs.

Build a secure platform

- ▶ IBM Worklight provides an extensible authentication model as part of its function. To comply with the Federal Information Processing Standards (FIPS), Hospital H1 uses Worklight with WebSphere Application Server for added protection. The hospital configures WebSphere Application Server to protect the application and adapters for the back-end servers and data.
- ▶ Using Worklight, Hospital H1 can grant access to data on a role, time, and location basis. Doctors can access patient records on mobile devices. However, it requires extra authentication approval if they are at home or on call to review the latest observations of patients. In addition, although doctors have access to the information of their patients, medical suppliers have access to check inventory and update stock.

Increase responsiveness and perceived value perception

- ▶ Hospital H1 is looking for a communication solution to find employees anywhere in the hospital. Using Worklight, the hospital can build an application that allows instant and secure communication. Doctors and nurses can quickly find colleagues without stopping what they are doing.
- ▶ Doctors at Hospital H1 must input prescriptions when their mobile devices are not connected to the network. JSONStore, the document-oriented storage system in Worklight, uses an encrypted container and ensures that the documents in the application are always available to doctors even when the devices running the application are offline.
- ▶ With the application, patients can pre-register for appointments and input their allergies and health history by using mobile devices. Worklight uses Secure Sockets Layer with server identity verification and enables communication over HTTPS to protect the information.

Reduce multi-platform development costs

- ▶ Worklight provides a standards-based platform and allows Hospital H1 to use third-party libraries and frameworks.
- ▶ Using Worklight, Hospital H1 can also create mobile applications quickly by using any combination of HTML5, native, and hybrid development methods.

Figure A-1 on page 256 shows the secured access from a mobile device to a back-end transactional core system on the Linux on IBM Z platform by using the global security policies and end-to-end secure transactions.

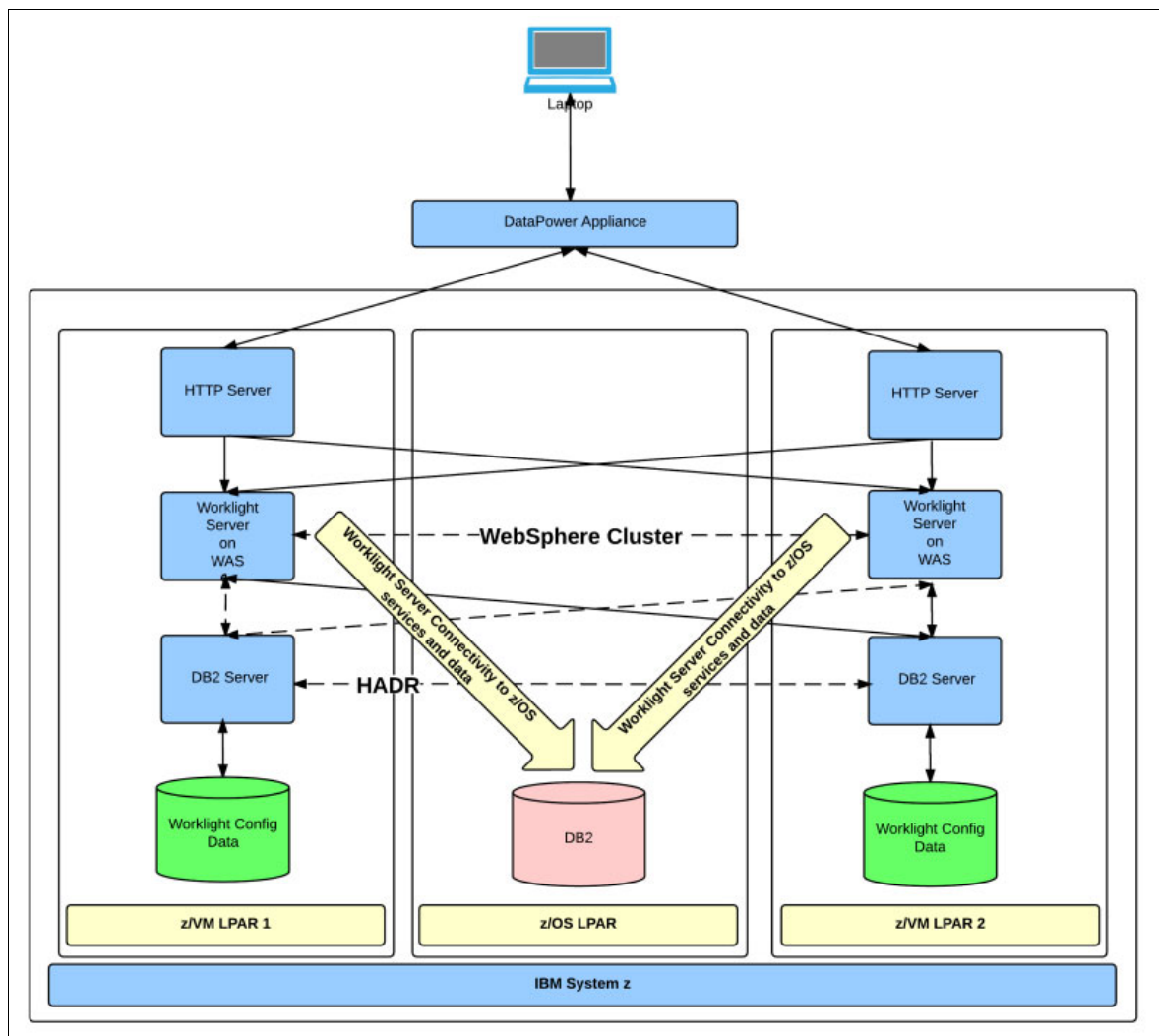


Figure A-1 Access from a mobile device to a back-end transactional core system

Financial Data Provider industry Hybrid Cloud Modernization

In this scenario, the fictional financial data provider, Fictional Bureau Company D1, selects the IBM Z platform for their on-premises cloud modernization platform. Bureau Company D1 wants to build a hybrid cloud platform, but they also want to reduce their cost of operation and overall data center footprint, as well as ensuring security and performance. The company's strategy is to deploy cloud native applications on a scalable and highly-available on-premises platform. Also, the company wants to have the option of deploying the applications on a public cloud provider and integrate deployment process through a continuous integration and continuous delivery (CI/CD) pipeline. In this example, the following consolidated hardware infrastructure can be employed:

- ▶ IBM Z z16
- ▶ IBM Z/VM 7.3
- ▶ Red Hat Enterprise Linux servers on IBM Z platform for building s390x images and integrating with CI/CD pipelines
- ▶ Red Hat OpenShift Container Platform
- ▶ IBM FlashSystem® 9500
- ▶ IBM Storage Fusion Data Foundation
- ▶ Corporate DNS
- ▶ Corporate Load Balancer

For a highly-available architecture, three z/VM LPARs were configured to host the Red Hat Enterprise Linux servers and the Red Hat OpenShift Container Platform nodes, so that a well distributed workload ensures resiliency and business continuity in case of hardware or software failures.

As the client had a stateful application, IBM Storage Fusion Data Foundation was used to provide a highly performing, highly available and secure environment for persistent storage on the Red Hat OpenShift Container Platform.

The CI/CD pipeline was developed to build the application images for both x86 and s390x, providing the client with the flexibility of deploying the applications on Linux on IBM Z on-premises and having the option of deploying applications on a public cloud.

Another important advantage of the employed architecture was the fact of the applications have strong affinity with Db2 on z/OS, which (through the concept of data gravity) attracts the workload to IBM Z. The tests comparing a Java application running on Linux on IBM Z and the public cloud showed important differences in performance in favor of Linux on IBM Z.

Figure A-2 on page 258 shows the added value of this solution and how it is scalable and integrated with automation, operation and administration.

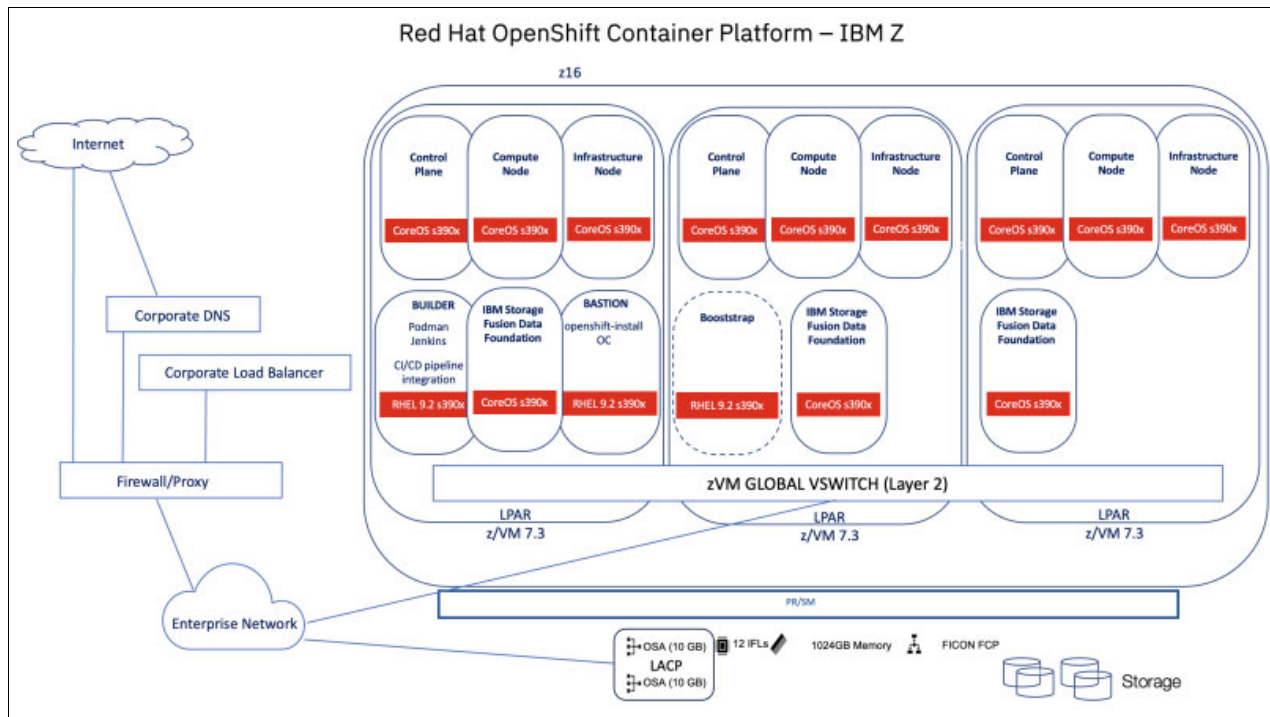


Figure A-2 Red Hat OpenShift Container Platform running on z/VM 7.3 on IBM Z

The Linux on IBM Z, z/VM hypervisor, and IBM Storage Fusion Data Foundation solution is an excellent productivity tool for any IT organization running Red Hat OpenShift Container Platform for microservices and cloud native applications.



B

z/VM Express System Installation

In this appendix we introduce the z/VM Express System Installation tool, and how it can help to simplify the migration of x86 workloads to IBM Z.

About z/VM Express System Installation (ESI)

z/VM ESI is a packaged hypervisor for IBM Z based on z/VM and a set of features and products pre-installed and customized. It is designed to make the initial setup of a hypervisor system for IBM Z simple and straightforward, requiring less system interaction and considerably less skill and experience than a standard installation of z/VM.

Packaged with z/VM ESI is a Linux system that provides a web interface and automated installation support for Red Hat OpenShift Container Platform and IBM Cloud Infrastructure Center. This Linux system is referred to as the Express Linux Automation and Networking guest, or ELAN.

Note: z/VM ESI is **not** an official product or offering from IBM. It is produced by a team of IBMers dedicated to improving the installation experience for newcomers to z/VM.

The only differences between a z/VM system installed in the “traditional” way and a z/VM ESI installation are the time saved using z/VM ESI and the benefit of decades of combined experience from those who have contributed to z/VM ESI.

How does z/VM ESI help?

Some of the ways that ESI can help are summarized in its name:

Express	Using ESI to install z/VM and products is quick and easy.
System	ESI provides an integrated system based on z/VM but provides much more than the base hypervisor alone.
Installation	ESI helps with the installation task not only for z/VM and its features and products but also for RHOCP and ICIC.

When setting up a hypervisor on IBM Z for migration of x86 workload, it can take a long time (several days or weeks) to install the base system and integrate all the components necessary to provide enterprise-grade virtualization capability. Using ESI, this time can be significantly reduced -- ESI makes it possible to deploy a brand new logical partition (LPAR) on an IBM Z server, install z/VM and all products, and then automatically provision a RHOCP cluster in under one hour.

What does z/VM ESI provide?

z/VM ESI was created by a team of IBM virtualization experts with experience installing z/VM in client environments. It provides an integrated hypervisor based on z/VM with the following features enabled:

- ▶ IBM Resource Access Control Facility (RACF)
- ▶ IBM Directory Maintenance Facility (DirMaint)
- ▶ IBM Performance Toolkit for z/VM (PerfKit)

In addition, the IBM Operations Manager for z/VM product is installed and configured with some helpful automation already enabled (such as automatic spool cleanup, automatic restart of critical system components, and console password suppression).

Finally, the Enterprise Linux Automation Network (ELAN) is available to provide the interface for performing deployment of RHOCP and/or ICIC. The ELAN provides a number of services in support of RHOCP, including DNS, load balancer, HTTP/S proxy, and container registry (for hosting internet disconnected or “airgap” installations of RHOCP and other products).

Using z/VM ESI to install a hypervisor system on IBM Z

The steps to install a system using z/VM ESI are the following:

- ▶ Complete the installation worksheet
- ▶ (Optional) save the worksheet values for automation
- ▶ Load the LPAR from the ESI media
- ▶ Provide the installation parameters to the installer
- ▶ Perform the installation
- ▶ Shut down the installer and IPL the installed system

Complete the installation worksheet

An installation of a system that will use z/VM ESI starts with completing a worksheet to capture all the details required for the installation. The worksheet captures details such as system name, DASD volumes, IP addresses, and so on.

A sample worksheet is shown in Table 9-1. For information regarding this worksheet, see the public documents, found at [about-zvmesi-public](#).

Table 9-1 Sample installation worksheet

z/VM Express System Install Planning Spreadsheet				
Fill in the fields as needed. See the documentation for field descriptions and valid values.				
Follow the instructions in the documentation for how to save this file.				
System Name (8 characters)	systemid			
System Group name (8 characters)	zvmesigroup			
Install type (optional)	selectname			
z/VM DASD (storage)				
Basic system DASD volumes	*	<u>Device Address (real device or UCB number)</u>	<u>Your label</u>	<u>Default label</u>
z/VM IPL volume (RES)	dasdsys1			M01RES
Common	dasdsys2			VMCOM1
VM Release	dasdsys3			730RL1
Additional Products	dasdsys4			M01U01
Spool	dasdsys5			M01S01
Dump (optional)	dasddump			M01S02
Linux volumes				
Label prefix	linxpx	ZVML		

Starting Address	linxfirst			
Number of Volumes	linxnum	1		
<u>z/VM Paging DASD</u>				
Label prefix	pagepfx	M01P		
Starting Address	pagefirst			
Number of Volumes	pagenum			
<u>Linux Cluster DASD</u>				
Label prefix	clstpfx	ZVMC		
Starting Address	clstfirst			
Number of Volumes	clstnum	6		
<u>z/VM Network</u>				
z/VM IP address	ipipaddr			
Gateway IP address	ipgateway			
Netmask (IP form or "/nn")	ipsnmask			
VLAN id (if defined)	ipqdiovlan			
OSA primary device address	ipdevnum			
OSA Port number (0 or 1)	ipqdioprtnum			
OSA failover device address	osasec			
OSA Port number (0 or 1)	osasecport			
MTU size	ipqdiomtu			
MAC address prefix (02xxxx)	macpfx			
z/VM Hostname	iphostname			
Domain (if not part of the host name)	ipdomain			
DNS Address 1 (required)	ipdnsaddr1			
DNS Address 2 (optional)	ipdnsaddr2			
DNS Address 3 (optional)	ipdnsaddr3			

Linux Network				
Linux IP address	Inxipaddr			
Host name (default name is shown)	Inxhostname	lxocpb01		
FTP restore server	*	(if used)		
Server address	ftpipaddr			
Login id	ftpuserid			
Password	ftppass			
Path and Directory	ftpdiname			

Save the worksheet values (optional)

z/VM ESI provides a method of automatically passing the values from the installation worksheet to the installer program in the system. This can save time and effort, and allow an installation to be performed completely through automation. This feature can be used by completing the worksheet using a supplied Excel spreadsheet (public documents, found at [about-zvemsi-public](#)) rather than a paper worksheet.

When the installation worksheet is completed and using the supplied Excel file, saving the details as a comma-separated-values (CSV) file produces the automation file. This CSV file is then transferred to the location where the rest of the z/VM ESI files are stored, ready for the system to use it. If the “automatic” flag is set in the file, the installer will validate the setting provided and, if okay, will proceed with the installation without interaction.

Load the LPAR from the ESI media

Using the standard functions on the IBM Z Hardware Management Console (HMC), the LPAR is first activated. Then the “Load from Removable Media or Server” function is used to load the z/VM ESI installer.

Note: If the automated (non-interactive) installation process using the CSV file is **not** being used, the Integrated 3270 Console for the LPAR needs to be opened prior to the Load function being initiated.

The Load from Removable Media or Server function can use either USB media connected to the HMC or a network-attached server. The standard protocol for this is FTP, although the HMC now also supports FTPS and SFTP (keys or certificates need to be set up in advance to use these secure protocols, though). If using a network server, the details of the server are entered on the panel. Clicking OK will make the HMC test the connection to the media and present the available loadable software.

Select the software to be loaded, and click OK. After one or two confirmation screens the HMC will commence loading the LPAR with the z/VM ESI installer.

Provide installation parameters to the installer

When the system starts operating after the load, it starts to read data from the supplied media files. It checks if a CSV setting file is present and reads any configuration values from the file.

If the automatic method is being used (using the “automatic” flag in the CSV file) the installer will validate the values read from the file and, if they are valid, will go ahead with the installation non-interactively. Output from the installation will be logged to the “Operating System Messages” function. Likewise, if there is an invalid setting in the CSV file the information will be displayed in Operating System Messages for diagnosis.

If an automatic installation was not requested (i.e. there is no “automatic” flag in the CSV file) IPL messages and instructions appear on the Integrated 3270 Console screen. The first thing to do is to configure the TCP/IP stack, which is done with the **SETUPNET** command. Any values that were provided in the CSV file are shown in the panel, but otherwise the fields have to be completed on screen. Once the values are set the information is saved and the system configures TCP/IP.

Note: Readers experienced with z/VM may be surprised that configuring TCP/IP is the first action! Not only does this provide immediate feedback that TCP/IP is functional, this is done so that there is an option to use FTP to transfer the remaining content for the installation. FTP is much faster than the interface between the LPAR and the HMC.

Once TCP/IP is configured the next step is to set the installation parameters for the system. This is done using the command **RESTORESIS**.

The system will prompt for the workload that will be supported for the installed system. This adjusts the subsequent panels to ask questions appropriate to the desired workload. For example, if IBM Cloud Infrastructure Center is selected then the system asks for at least 150GB of disk space for Linux, and if RHOCIP is selected then a separate disk pool with larger disks is requested.

Once the workload type is selected, the system looks for FCP adapters. If FCP adapters are present on the LPAR, the system prompts to scan for SCSI LUNs and define them as z/VM Emulated Devices (EDEVs) for use in the subsequent steps.

Once FCP devices are defined (or if there are no FCP devices) the main **RESTORESIS** panel is displayed. Again, any values read from the CSV will be filled in on the screen, otherwise values must be manually provided.

For filling in device addresses, the program provides a prompt capability. Pressing **PF4** pops up a selection box from which the device address or addresses to be used can be selected.

Once values are completed, they are saved using **PF5**.

Perform the installation

As mentioned, if the “automatic” installation method was requested the installation would automatically commence once the installation values were validated.

For an interactive installation, the install is started using the command **RESTORESIS START**. The progress of the installation appears in the Integrated 3270 Console screen.

As part of the installation process, the new z/VM ESI system is IPLed at second-level under the installation system. This tests that the install has been successful.

Once the installation process is complete, the system provides instructions on how to restart the LPAR and use the new z/VM ESI system.

Shut down the installer and IPL the installed system

Using the SHUTDOWN command, the installer is stopped. Then, the **Load** function on the HMC is used to start the LPAR with the installed z/VM system. The appropriate type of IPL (depending on the type of disk used) is selected, and the parameters entered.

When the load is performed for the first time, the final configuration of the system is completed. Importantly, the supplied Linux guest has the IP address configuration applied to it, and certificates for the Linux and z/VM secure interfaces are regenerated.

Using the ELAN to install ICIC

The web interface of the ELAN provides the way to perform an automated installation of ICIC. The ELAN supports adding additional components to your system in the form of “modules”, which can be downloaded from the location where z/VM ESI was obtained. The ICIC module contains all the requirements to perform an automated installation of ICIC.

Having downloaded the ICIC module, the first step is to upload the module to the ELAN. This is done either by using SSH to copy the module file to the `/opt/content` directory, or by using the ESI Content Modules page in the ELAN web interface to upload using the web browser.

Once the module is present on the ELAN, the page “IBM Cloud Infrastructure Center” is used to initiate the automated installation. All of the ICIC modules found on the ELAN are listed on the page.

Note: The terms of the ICIC license require that only the latest version be installed. It is your installation’s responsibility to comply with the terms of the license agreement.

The page on the ELAN will change if multiple ICIC modules are found. The latest version will be highlighted, but other versions will be available. A message similar to the above paragraph (a reminder to install the latest version) is also shown.

To initiate the installation, click the “Deploy” button next to the appropriate ICIC module entry in the list. The installation, managed by scripts and Ansible playbooks, will be performed in the background.

When the automation to install ICIC is complete, the page will change to show a link to the ICIC web interface.

Related publications

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

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in soft copy only.

- ▶ *Advanced Networking Concepts Applied Using Linux on IBM System z*, SG24-7995
- ▶ *An Introduction to z/VM Single System Image (SSI) and Live Guest Relocation (LGR)*, SG24-8006
- ▶ *DB2 10 for Linux on System z Using z/VM v6.2, Single System Image Clusters and Live Guest Relocation*, SG24-8036
- ▶ *Experiences with Oracle Database 12c Release 1 on Linux on System z*, SG24-8159
- ▶ *Experiences with Oracle Solutions on Linux for IBM System z*, SG24-7634
- ▶ *IBM z16 A02 and IBM z16 AGZ Technical Guide*, SG24-8952
- ▶ *IBM Z Connectivity Handbook*, SG24-5444
- ▶ *IBM System Storage SAN Volume Controller Best Practices and Performance Guidelines*, SG24-7521
- ▶ *Implementing FlashSystem 840 with SAN Volume Controller*, TIPS1137
- ▶ *Implementing the IBM System Storage SAN Volume Controller V7.2*, SG24-7933
- ▶ *Introduction to the New Mainframe: z/VM Basics*, SG24-7316
- ▶ *Introduction to Storage Area Networks and System Networking*, SG24-5470
- ▶ *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926
- ▶ *Security for Linux on System z*, SG24-7728
- ▶ *Security on z/VM*, SG24-7471
- ▶ *Set up Linux on IBM System z for Production*, SG24-8137
- ▶ *The Virtualization Cookbook for IBM Z Volume 1: IBM z/VM 7.2*, SG24-8147
- ▶ *The Virtualization Cookbook for IBM Z Volume 2: Red Hat Enterprise Linux 8.2*, SG24-8303
- ▶ *The Virtualization Cookbook for IBM z Systems Volume 3: SUSE Linux Enterprise Server 12*, SG24-8890
- ▶ *The Virtualization Cookbook for IBM z Systems Volume 4: Ubuntu Server 16.04*, SG24-8354
- ▶ *Virtualization Cookbook for IBM Z Volume 5: KVM*, SG24-8463
- ▶ *Linux on IBM eServer zSeries and S/390: Application Development*, SG24-6807

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

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- Discover servers, storage and software designed for your enterprise hybrid cloud and AI strategy

<https://www.ibm.com/it-infrastructure>

- GNU assembler manual

<http://www.gnu.org/software/binutils>

- IBM Z and Cloud Modernization Center

<http://www-03.ibm.com/systems/services/labservices/solutions/hacoc.html>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



Practical Migration from x86 to Linux on IBM Z

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



Practical Migration from x86 to Linux on IBM Z

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



Practical Migration from x86 to Linux on IBM Z

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



Practical Migration from x86 to Linux on IBM Z

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

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



Redbooks

Practical Migration from x86 to Linux on IBM Z

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



Redbooks

Practical Migration from x86 to Linux on IBM Z

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



Practical Migration from x86 to Linux on IBM Z

**A guide to migrating
popular applications
and services from
Linux on x86 to Linux
on IBM Z**

**Practical guidance on
planning, analysis,
and TCO**

**Comprehensive
hands-on migration
case study**

There are many reasons why you would want to optimize your servers through virtualization using Linux on IBM Z:

- ▶ Too many distributed physical servers with low utilization
- ▶ A lengthy provisioning process that delays the implementation of new applications
- ▶ Limitations in data center power and floor space
- ▶ High total cost of ownership (TCO)
- ▶ Difficulty allocating processing power for a dynamic environment

This IBM Redbooks publication provides a technical planning guide and example for IT organizations to migrate from their x86 environment to Linux on IBM Z. It begins by examining the benefits of migrating workloads to Linux on IBM Z. Here, we describe the workload centric method of information technology and then discuss the benefits of migrating workloads to Linux on IBM Z.

Next, we describe total cost of ownership analyses and we guide you in understanding how to analyze your environment before beginning a migration project. We also assist you in determining the expected consolidation ratio for a given workload type.

We also describe virtualization concepts along with describing the benefits of migrating from the x86 environment to guests residing on an IBM z/VM single system image with live guest relocation.

This IBM Redbooks publication walks you through a migration approach, includes planning worksheets, as well as a chapter to assist you in analyzing your own systems. We also discuss postmigration considerations such as acceptance testing of functionality and performance measurements.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks