

Practical Migration from x86 to LinuxONE

Lydia Parziale

Eric R. Farman

Manoj S. Pattabhiraman



Linux



International Technical Support Organization

Practical Migration from x86 to LinuxONE

January 2017

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

First Edition (January 2017)

This edition to LinuxONE on IBM Rockhopper and IBM Emperor.

© Copyright International Business Machines Corporation 2017. 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	vii
Trademarks	viii
Preface	ix
Authors	x
Now you can become a published author, too!	x
Comments welcome	xi
Stay connected to IBM Redbooks	xi
Part 1. Decision-making	1
Chapter 1. Benefits of migrating workloads to LinuxONE	3
1.1 Benefits	4
1.2 Reasons to choose LinuxONE	4
1.2.1 Best of Enterprise Linux and Open Source	5
1.2.2 Hardware strengths	5
1.3 A new type of information technology: Workload centric	7
1.4 Workload centric cloud	8
1.5 Enterprise cloud computing blueprint for IBM LinuxONE	11
1.5.1 Cloud Solutions on IBM LinuxONE	11
Chapter 2. Analyze and understand	13
2.1 Total cost of ownership analysis	14
2.2 Migration planning	14
2.3 Choosing workloads to migrate	15
2.4 Analysis of how to size workloads for migration	15
2.5 Financial benefits of a migration	16
2.6 Project definition	18
2.7 Planning checklists	18
2.7.1 Product and tools checklist	18
2.7.2 Application implementation checklist	19
2.7.3 Application environment checklist	20
2.7.4 Training checklist	21
2.7.5 Hardware planning checklist	21
Chapter 3. Virtualization concepts	23
3.1 The demand for virtualization	24
3.2 Typical x86 virtualization	24
3.3 LinuxONE virtualization	25
3.3.1 Process Resource/System Manager hypervisor	26
3.3.2 Dynamic Partition Manager	26
3.3.3 KVM hypervisor	26
3.3.4 z/VM hypervisor	27
3.4 Linux guest	28
3.5 Guest mobility	30
3.5.1 KVM guest migration	30
3.5.2 z/VM single system image and live guest relocation	30
3.6 KVM hypervisor components	32
3.6.1 Linux kernel and KVM module	32

3.6.2	QEMU	32
3.6.3	The libvirt API	33
3.6.4	Resource management.	33
3.7	z/VM hypervisor components	33
3.7.1	Control program	33
3.7.2	Conversational Monitor System	34
3.7.3	IBM Wave	34
3.8	Virtualized resources.	35
3.8.1	Virtualized CPU.	35
3.8.2	Virtualized memory	35
3.8.3	Virtualized disk	38
3.8.4	Virtualized network	39
Part 2.	Migration	41
Chapter 4.	Migration process	43
4.1	Stakeholder definitions	44
4.1.1	Business stakeholders	44
4.1.2	Operational stakeholders	45
4.1.3	Security stakeholders	47
4.2	Identify the stakeholders	47
4.3	Assembling the stakeholders	48
4.3.1	Communicating the change	48
4.4	Migration methodology	49
4.4.1	Pre-assessment	49
4.4.2	Define success criteria	50
4.4.3	Finalize the new environment	50
4.4.4	Pilot proof of concept	51
4.4.5	Decision to migrate	51
4.4.6	Resource estimation	51
4.4.7	Actual migration	52
4.4.8	Verification testing.	52
4.4.9	Check against success criteria	53
Chapter 5.	Migration analysis	55
5.1	Network analysis	56
5.1.1	Network facilities available on LinuxONE and KVM	56
5.1.2	Network facilities available on LinuxONE and z/VM	57
5.1.3	Network migration overview	57
5.1.4	Helpful steps for a network migration	66
5.2	Storage analysis	66
5.2.1	Data migration.	66
5.2.2	LinuxONE: pre-installation considerations	70
5.3	Application analysis.	76
5.3.1	Why migrate applications	76
5.3.2	Which applications can be migrated	77
5.3.3	Selecting an application for migration to LinuxONE	77
5.3.4	Applications that are best suited for migration	78
5.3.5	Other software	79
5.3.6	Selecting an application for a proof of concept	80
5.3.7	Application interdependencies	81
5.3.8	Successful application migration.	81
5.3.9	Special considerations for migrating a Java application	81
5.3.10	Special considerations for migrating C++ applications	82

5.3.11	Middleware, libraries, and databases	83
5.3.12	Helpful steps for an application migration	83
5.4	Database analysis	84
5.4.1	Before database migration	84
5.4.2	Migrating a single instance	84
5.4.3	Migrating multiple instances	84
5.4.4	Technical considerations	86
5.4.5	Migrating DB2 and Oracle from x86 to LinuxONE	89
5.4.6	Tips for successful migration.	90
5.5	Backup analysis	91
5.5.1	Introduction to backup and archival concepts.	91
5.5.2	KVM backup	92
5.5.3	z/VM backup	92
5.5.4	Linux backup	93
5.5.5	Migrating backed-up and archived data	93
5.5.6	General archival migration considerations	94
5.5.7	Migrating to new backup software.	94
5.6	Security analysis	95
5.6.1	Security migration overview	95
5.6.2	Code and application analysis	99
5.6.3	Availability and accountability	101
5.6.4	Data integrity and confidentiality	102
5.6.5	Security change management	104
5.6.6	Enterprise authentication options	104
5.6.7	CP Assist for Cryptographic Function	104
5.7	Operational analysis	105
5.7.1	Operational migration tasks	105
5.8	Disaster recovery and availability analysis	107
5.8.1	Availability analysis	107
5.8.2	Single points of failure.	108
5.8.3	LinuxONE features for high availability	108
5.8.4	Availability scenarios.	109
5.8.5	Linux-HA Project	115
5.8.6	High Availability add-ons.	115
5.8.7	Understanding the availability requirements of your applications	116
5.9	Virtualized Environment to LinuxONE Cloud Migration.	117
5.9.1	Hypervisor Considerations	117
5.9.2	Network considerations.	118
5.9.3	Security considerations.	118
5.9.4	LinuxONE cloud management	119
	Chapter 6. Hands-on migration	121
6.1	Setting up the system	122
6.1.1	Software products and tools checklist.	122
6.1.2	Hardware checklist	122
6.1.3	FCP and multipath	123
6.1.4	FCP migration setup tasks	124
6.2	Migrating DB2 and its data	125
6.3	Migrating the WebSphere Application Server	127
	Chapter 7. Post migration considerations	129
7.1	Gaining acceptance	130
7.2	Performance measurement.	130

7.2.1	What is performance	131
7.2.2	Choosing what to measure	131
7.3	Performance tuning	132
Part 3.	Deployment	135
Chapter 8.	Deployment of workloads	137
8.1	Deciding between containers and virtual machines	138
8.2	Setting up Docker on Ubuntu 16.04 LTS	139
8.3	Deploying MongoDB on IBM LinuxONE	141
8.3.1	Work environment	141
8.3.2	MongoDB container deployment	142
8.3.3	Running MongoDB container	144
8.3.4	Verifying and accessing MongoDB container	144
8.4	Deploying Hyperledger (Blockchain) Fabric on LinuxONE	145
8.4.1	Environment	145
8.5	Deploying high availability clustering	152
8.6	Deploying MediaWiki and MySQL	152
8.6.1	Analysis and planning	153
8.6.2	Installing the LAMP stack	153
8.6.3	Starting and testing LAMP components	154
8.6.4	Migrating iSCSI disks containing MySQL and MediaWiki	159
8.7	Deploying OpenLDAP	165
8.7.1	Analysis and planning	165
8.7.2	Installing LDAP software	166
8.7.3	Configuring the OpenLDAP service	166
8.7.4	Export OpenLDAP data from x86 server	170
8.7.5	Import OpenLDAP data to LinuxONE	171
8.7.6	Verify OpenLDAP is working	171
8.8	Deploying central log server	172
8.8.1	Analysis and planning	172
8.8.2	Initial configuration	172
8.8.3	Server configuration	174
8.8.4	Client configuration	175
8.8.5	Testing syslog-ng	176
8.8.6	Migrating using syslog-ng	176
8.9	Deploying Samba	176
8.9.1	Installing Samba software	177
8.9.2	Configuring Samba	177
Part 4.	Appendix	181
Appendix A.	Additional use case scenarios	183
Telecom industry consolidation and cloud		184
Healthcare industry: Mobile and internet solution		185
Energy and utilities industry: SAP Cloud and Automation solution on System z		187
Related publications		191
IBM Redbooks		191
Online resources		192
Help from IBM		192
Index		193

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](http://www.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 LinuxONE Rockhopper™	ProtecTIER®
BigInsights®	IBM SmartCloud®	Redbooks®
DB2®	IBM Spectrum™	Redbooks (logo)  ®
DB2 Connect™	IBM Spectrum Protect™	System Storage®
DS8000®	IBM Spectrum Scale™	System z®
ECKD™	IBM z Systems®	Tivoli®
FICON®	InfoSphere®	WebSphere®
FlashCopy®	OMEGAMON®	z Systems®
GDPS®	Parallel Sysplex®	z/OS®
HiperSockets™	POWER®	z/VM®
IBM®	PR/SM™	zEnterprise®
IBM LinuxONE™	Processor Resource/Systems	
IBM LinuxONE Emperor™	Manager™	

The following terms are trademarks of other companies:

Inc., and Inc. device are trademarks or registered trademarks of Kenexa, an IBM Company.

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.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

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

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

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

Preface

LinuxONE is a portfolio of hardware, software, and solutions for an enterprise-grade Linux environment. It has been designed to run more transactions faster and with more security and reliability specifically for the open community. It fully embraces open source-based technology.

Two servers are available for LinuxONE: The LinuxONE Emperor and Rockhopper.

The IBM® LinuxONE Emperor server has these features:

- ▶ Near limitless scaling and support of up to 8,000 virtual Linux servers on a single footprint
- ▶ Intrinsic platform security, and safer and faster data encryption than x86
- ▶ Faster critical workloads and near real-time insights with higher performance and throughput than x86 with a net lower cost per transaction
- ▶ Lowered costs by managing fewer servers, reducing complexity, gaining unparalleled utilization and applying usage-based pricing

The IBM LinuxONE™ Rockhopper server has these features:

- ▶ Unprecedented performance, security, and resiliency
- ▶ Scale up and out with up to one million Docker containers per system
- ▶ Manage more transactions more quickly with 34 percent greater capacity
- ▶ Gain faster response times with massive high-performance I/O throughput
- ▶ Faster insights with large memory configurations that are optimized for Linux
- ▶ Delivery of data and services on the world's most secure Linux platform

Aside from still running SUSE Linux Enterprise Server and Red Hat Enterprise Linux Servers, LinuxONE runs Ubuntu, which is popular on x86 hardware.

Ubuntu, which runs the cloud, smartphones, a computer that can remote control a planetary rover for NASA, many market-leading companies, and the Internet of Things, is now available on IBM LinuxONE servers. Together, these two technology communities deliver the perfect environment for cloud and DevOps. Ubuntu 16.04 on LinuxONE offers developers, enterprises, and Cloud Service Providers a scalable and secure platform for next generation applications that include OpenStack, KVM, Docker, and JuJu.

The following are reasons why you would want to optimize your servers through virtualization using LinuxONE:

- ▶ 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 reference for IT organizations that are considering a migration from their x86 distributed servers to LinuxONE. This book walks you through some of the important considerations and planning issues that you might encounter during a migration project. Within the context of a pre-existing UNIX based or x86 environment, it presents an end-to-end view of the technical challenges and methods necessary to complete a successful migration to LinuxONE.

Authors

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

Lydia Parziale is a Project Leader for the ITSO team in Poughkeepsie, New York, with world-wide experience in technology management including software development, project leadership, and strategic planning. Her areas of expertise include business development and database management technologies. Lydia is a certified PMP and an IBM Certified IT Specialist with an MBA in Technology Management, and has been employed by IBM for 25+ years in various technology areas.

Eric R. Farman is a Senior Software Engineer for IBM z Systems® Virtualization Development in Endicott, New York, with 10 years of experience with IBM z/VM® development and four years with KVM development. He holds a degree in Computer Science from Susquehanna University. His areas of expertise include device connectivity and operating system exploitation of virtualized devices. He has presented at numerous technical conferences on various topics.

Manoj S. Pattabhiraman is an Open Source Solutions Architect from the Asia Pacific - IBM z Systems Solutions Team. He has more than 16 years of experience in Open Source Solutions (virtualization, cloud, and Blockchain). In his current role, he drives and provides consultation and implementation services for various Linux customers across Asia Pacific region. Manoj has contributed to several z/VM and LinuxONE publications, and has been a frequent presenter at various technical conferences and workshops on Cloud, Blockchain, and Virtualization.

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

Helene Bastide-Grosch
Tristan Spadi
Eric Phan
IBM Systems, Montpellier Client Center

Brett Webb
Richard Young
IBM USA

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

- ▶ Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- ▶ Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- ▶ 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>



Part 1

Decision-making

This part describes the key benefits of migrating to LinuxONE and the reasons to migrate to LinuxONE. It also outlines some of the considerations when deciding to migrate to LinuxONE. The part concludes with some virtualization concepts to help you to understand and compare x86 virtualization, and both KVM and IBM z/VM virtualization.

This part includes the following chapters:

- ▶ Chapter 1, “Benefits of migrating workloads to LinuxONE” on page 3
- ▶ Chapter 2, “Analyze and understand” on page 13
- ▶ Chapter 3, “Virtualization concepts” on page 23



Benefits of migrating workloads to LinuxONE

This section describes the benefits and reasons to migrate workloads to LinuxONE. Additionally, it describes a new type of information technology that is called *workload centric* and explains how the workload centric cloud benefits from LinuxONE with an enterprise cloud computing blueprint.

This chapter includes the following sections:

- ▶ Benefits
- ▶ Reasons to choose LinuxONE
- ▶ A new type of information technology: Workload centric
- ▶ Workload centric cloud
- ▶ Enterprise cloud computing blueprint for IBM LinuxONE

1.1 Benefits

Linux is available on a large variety of computing platforms from set top boxes and handheld devices to the largest servers. This flexibility 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 supports your application. Workloads running on LinuxONE benefits from a hardware platform that includes specialized processors, cryptographic cards with dedicated RISC processors, and a combination of hypervisors that allow unparalleled flexibility in Linux deployment.

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. Hundreds of Linux distributions are available for almost every computing platform. The following are the three enterprise distributions¹ of Linux:

- ▶ Red Hat: Red Hat Enterprise Linux (RHEL)
<http://www.redhat.com>
- ▶ SUSE: SUSE Linux Enterprise Server
<http://www.suse.com>
- ▶ Canonical: Ubuntu Server
<http://www.ubuntu.com>

All of these Linux distributions provide customers who use Linux with various support options, including 24 x 7 support with one-hour response time worldwide for customers running production systems. In addition to the Linux operating system, all the major Linux distributions 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 the distributors to resolve the problem.

The increased interest and usage 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 means significantly lower support costs and, for 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 for the ability to scale databases across highly scalable hardware.

1.2 Reasons to choose LinuxONE

IBM LinuxONE 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.

IBM LinuxONE provides the highest levels of availability (near 100 percent uptime with no single point of failure), performance, throughput, and security. End-to-end security is built in

¹ A Linux distribution is a complete operating system and environment. It includes 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.

with isolation at each level in the stack, and provides the highest level of certified security in the industry.

Additionally, LinuxONE Systems facilitate transparent use of redundant processor execution steps and integrity checking, which is necessary in financial services industries. LinuxONE 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.

IBM LinuxONE delivers on the promise of a flexible, secure, and smart IT architecture that can be managed seamlessly to meet the requirements of today's fast-changing business climate.

1.2.1 Best of Enterprise Linux and Open Source

LinuxONE provides these 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 of running Linux on industry's most secure and resilient hardware platform.
- ▶ Easy integration of data and applications with existing IBM z Systems based solutions.
- ▶ Overall increases the operational IT efficiency.

1.2.2 Hardware strengths

IBM LinuxONE provides these hardware strengths:

- ▶ 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 by using IBM Processor Resource/Systems Manager™ (IBM PR/SM™)².
 - Up to 85 (LinuxONE Emperor) or 40 (LinuxONE Rockhopper) logical partitions (LPAR) with independent virtual resources.

² PR/SM is standard component of all IBM LinuxONE models, which enables LPARs to share system resources. PR/SM divides physical system resources, both dedicated and shared, into isolated logical partitions. Each partition is like an independent system running its own operating environment. It is possible to add and delete resources like processors, I/O, and memory across partitions while they are actively in use.

- PR/SM is one of the most secure systems available, having achieved Common Criteria Evaluation Assurance Level 5+ (EAL5+) for partition isolation. This is one of the highest levels of certification 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 following site:

<http://www.commoncriteriaportal.org>

The certified evaluation levels for IBM z Systems Operating Systems are available as of March 2015:

- ▶ IBM z/VM version 6 release 3: Certified at EAL4+
- ▶ Red Hat Enterprise Linux 7: Certified at EAL4+
- ▶ SUSE Linux Enterprise Server 12: Certified at EAL4+

- IBM Dynamic Partition Manager provides facilities to define and run virtualized computing systems by using a firmware managed environment that coordinates the physical system resources shared by the partitions. The partitions resources include processors, memory, network, storage, crypto, and accelerators.
- Both the industry-leading virtualization hypervisor z/VM and the open source hypervisor kernel-based virtual machine (KVM) are supported on all IBM LinuxONE models.
- PR/SM, z/VM, and KVM employ hardware and firmware innovations that make virtualization part of the basic fabric of the IBM LinuxONE platform.
- IBM HiperSockets™ allows up to 32 virtual LANs, thus allowing memory-to-memory TCP/IP communication between partitions.
- ▶ Scalability:
 - IBM LinuxONE Emperor™ scales to 141 physical processors and up to 10 TB of memory.
 - IBM LinuxONE Rockhopper™ scales to 20 physical processors and up to 4 TB of memory.
- ▶ Security:
 - Clear key integrated cryptographic functions provide high-speed cryptography for data in memory. These functions Support Data Encryption Standard (DES) and Triple Data Encryption Standard (TDES), Secure Hash Algorithm (SHA) for up to 512 bits, Advanced Encryption Standard (AES) for up to 256 bits, and Pseudo Random Number Generation (PRNG).
 - 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 140-2 level 4.
- ▶ Just-in-time deployment of resources:
 - On/Off Capacity on Demand provides temporary processing capacity to meet short-term requirements or for testing new applications.
 - 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 an IBM LinuxONE 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.

► Power and cooling savings:

With its low power and cooling requirements, IBM LinuxONE is an ideal platform for the consolidation of distributed servers.

1.3 A new type of information technology: Workload centric

An IT workload can be described as one or more system resources that are used by one system or a combination of systems to complete one or more jobs at the same time or time period using system resources such as CPU, memory, and I/O.

This definition might seem a little simple. However, if you try to create boundaries and focus only on one workload on one system, each workload has its own characteristics and its own behaviors, so it is not easy to separate system functionality from infrastructure.

A classic IT workload approach such as “workload silos” no longer meets the new business and IT requirements such as workload deployments for analytics, big data, or secure commerce solutions on multiple components. This limitation is especially true now that cloud has been positioned as an answer to all IT requirements.

Currently, when deploying a workload, multiple components need to be wired together: Application code, middleware, management agents, preexisting services, virtual machines, data, disks, and networks. Workload availability and performance depend on the correct wiring as shown in Figure 1-1. The question becomes how to provision and maintain the components that are used by a cloud workload to achieve both agility and optimization for a varying set of workloads.

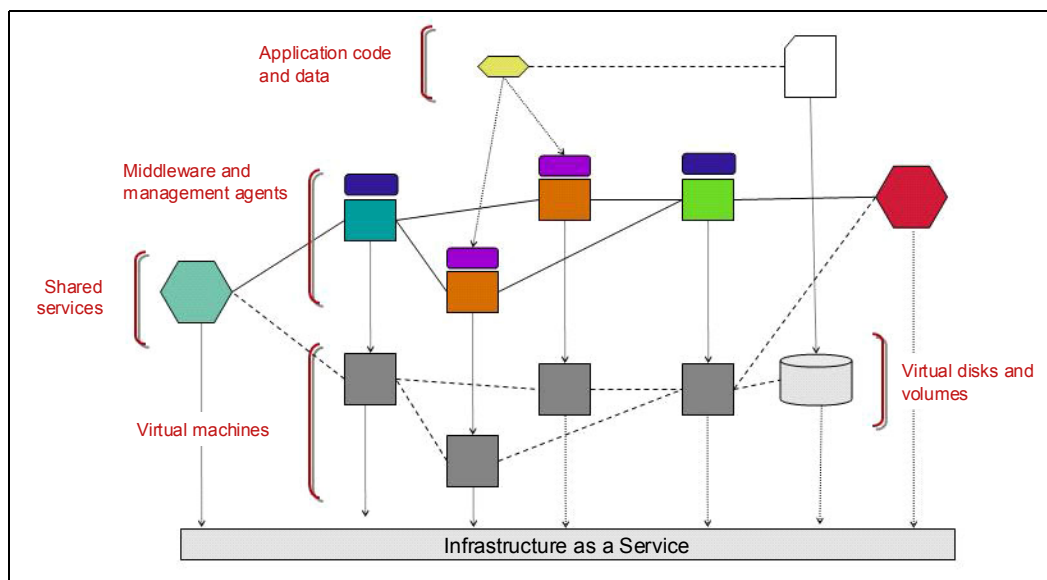


Figure 1-1 What is workload

The answer is to understand your dynamic workload patterns and dynamic automation composition for your heterogeneous system as well as autonomic and proactive management such as that shown in Figure 1-2.

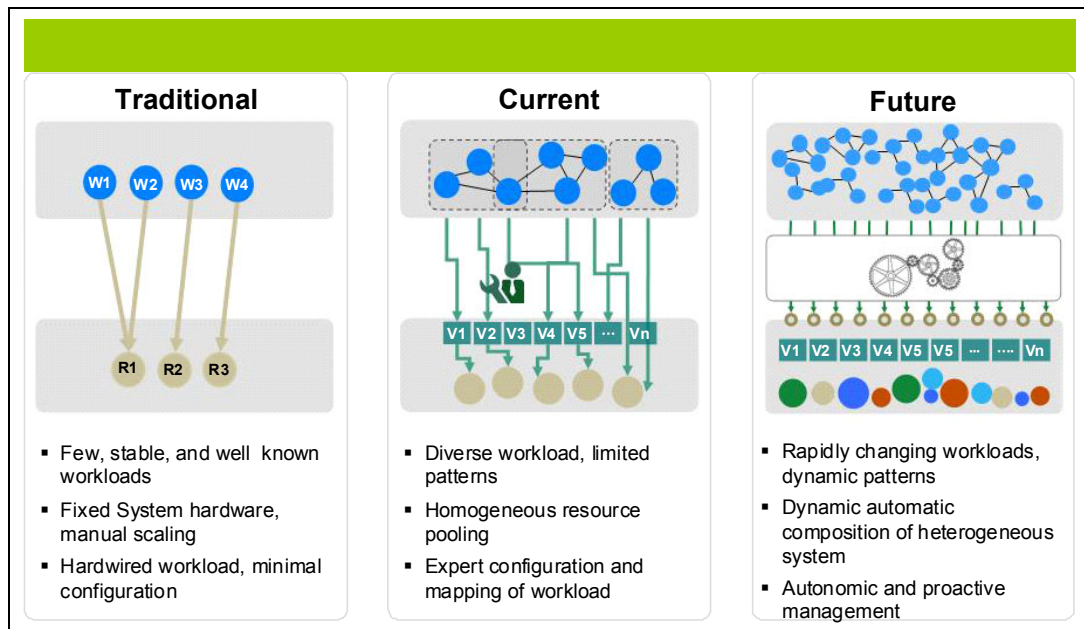


Figure 1-2 Future of workload management

The future, as shown in Figure 1-2, can be considered as the intersection of three main areas:

- ▶ Consumability
- ▶ Agility
- ▶ Efficiency

The intersection of these three main areas requires reconsideration of how to design an IT workload, services, and infrastructure cloud solution. A Workload centric Cloud provides abstraction and solution of workloads, services, and infrastructure, and an end-to-end mapping of an intelligent software-defined system environment.

1.4 Workload centric cloud

A workload centric cloud can be described as a workload aware cloud solution. However, this awareness is not covered by existing software as a service (SaaS), platform as a service (PaaS), or infrastructure as a service (IaaS) solutions because these systems run “cloud enabled” workloads on their scalable, virtualized heterogeneous infrastructure. They are not aware of the often varying workload definitions. They can only be aware of when system resources are busy or which system is running which workload. This section describes a different cloud architecture: A workload centric cloud.

A workload centric cloud offers a programmable and optimized way of continuously delivering and running workload.

As you can see in Figure 1-3, this architecture is beyond what is commonly known as a *cloud architecture*. It is called a software-defined environment, and it can help make the data center more customized and efficient by using the diverse infrastructure according to workload types, business rules, and resource availability. The entire IT infrastructure is controlled not manually and by hardware, but by software and workloads that are serviced automatically by the most appropriate resource.

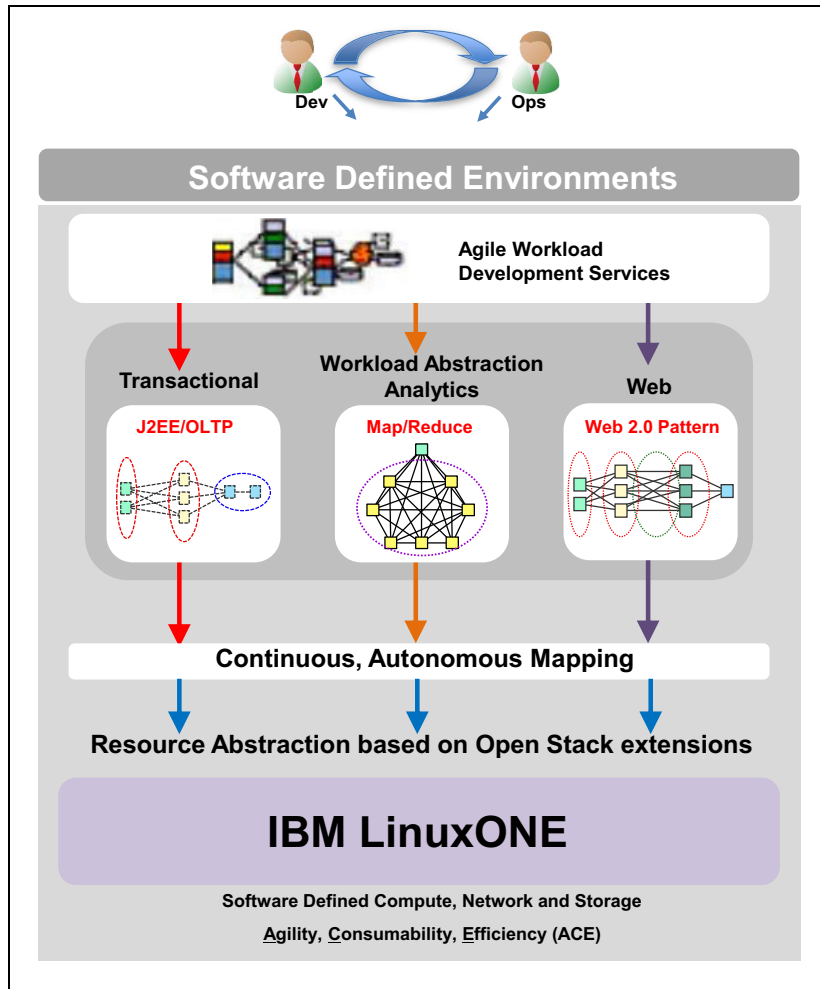


Figure 1-3 Workload centric cloud architecture

This architecture must be smarter, able to analyze and learn workloads, and be able to optimally manage everything.

One architecture to achieve a software-defined environment is shown in Figure 1-4. In this solution, workloads are defined and orchestrated by using patterns and resources that are managed and deployed according to business rules and policies.

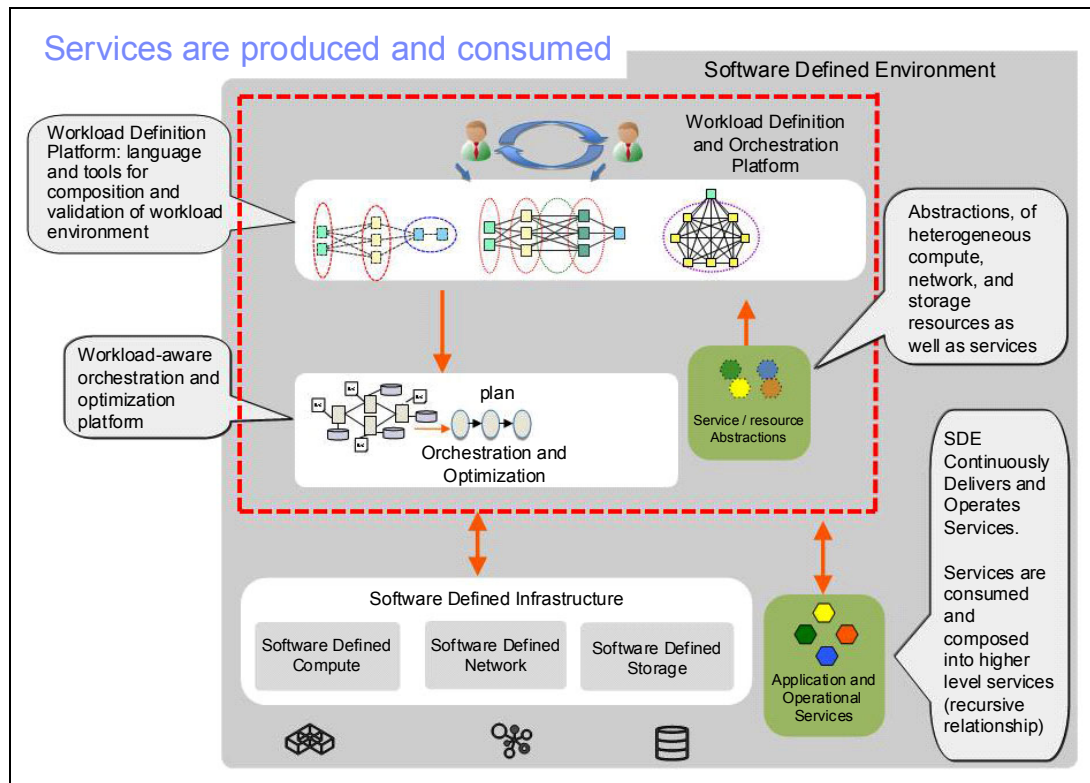


Figure 1-4 Software-defined environment for workload centric cloud

In this architecture, the workload definitions and resource abstraction are made known to the orchestration platform. Then business rules that are used to identify required service components are defined along with the necessary infrastructure resources.

The infrastructure resources are then managed by these components:

- ▶ Software-defined compute provides workload-aware infrastructure management and optimization.
- ▶ Software-defined networks for virtual environments, such as Linux, create a virtual network for virtual machines that is decoupled and isolated from the physical network, which can often become a bottleneck.
- ▶ Software-defined storage allows for the management of huge surges in data driven by mobile and social technologies. It does so by pooling physical storage resources, regardless of the vendor, to improve utilization in a cost-effective manner.

1.5 Enterprise cloud computing blueprint for IBM LinuxONE

Enterprise computing encompasses all the various types of business software solutions including database management software such as IBM DB2®.

Enterprise cloud computing refers to the computer environment that delivers infrastructure, such as web services, software, and platform services, to your entire enterprise. Figure 1-5 shows the IBM LinuxONE platform ready for each layer of cloud.

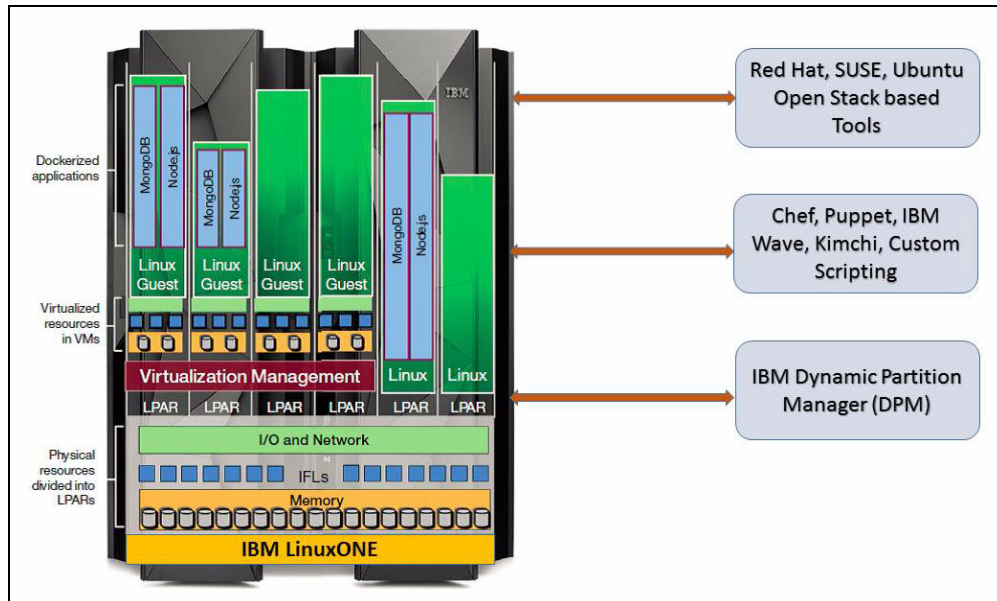


Figure 1-5 IBM LinuxONE is “cloud ready”

1.5.1 Cloud Solutions on IBM LinuxONE

Cloud is a method for delivering computing services that allows users to self-service provision environments with the agility that their businesses need. Systems running mission-critical applications are the backbone of an enterprise. When they go down, there is major impact to a business. Because of these factors, these workloads require a cloud solution that has exceptional system uptime, excellent data security/privacy, and a powerful vertical scale architecture. For these work-loads, IBM LinuxONE is the most securable, reliable, and scalable on-premises cloud solution. LinuxONE can provide the same agility and time to value as other cloud services, along with unparalleled enterprise qualities of service.

IBM LinuxONE allow those delivering cloud services to rapidly deploy a trusted, scalable OpenStack-based Linux cloud environment that can start small and scale up to 6000 virtual machines in a single footprint on IBM LinuxONE.

Virtualization portfolio

Establishing cloud environments on IBM LinuxONE 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 these applications for infrastructure and virtualization management:

- ▶ IBM LinuxONE Hardware: IBM Emperor, IBM Rockhopper:
 - Massively scalable.
 - Characterized by excellent economics and efficiencies.
 - Highly secure and available.
- ▶ z/VM 6.3:
 - Support more virtual servers than any other platform in a single footprint.
 - Integrated OpenStack support.
- ▶ KVM v1.1 for IBM LinuxONE:
 - Provides a choice for clients who want open virtualization while taking advantage of the robustness, scalability, and security of the LinuxONE platform.
 - The standard interfaces that it provides allows for easy integration into an existing infrastructure.
- ▶ Linux for IBM LinuxONE:
 - Distributions 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.

Cloud Solutions on IBM LinuxONE

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

For z/VM, the Cloud Manager Appliance (CMA) provides an easy method to deploy z/VM OpenStack enablement. OpenStack products and solutions can be constructed to use as many or as few of the services as is appropriate. That might mean that the CMA runs cloud controller services, compute node services, or only services needed by OpenStack z/VM drivers running in other virtual machines or on other platforms.

IBM LinuxONE provides the following standardization and automation advantages with Extreme Cloud Administration Toolkit (xCAT) on z/VM:

- ▶ Included with z/VM 6.3
- ▶ Allows customers to set up a rudimentary cloud environment, without acquiring any additional products, based on open source code
- ▶ Image-based cloud service delivery with integrated provisioning, monitoring, service catalog and service desk, storage management, and high-availability
- ▶ Immediate monitoring for IBM z/VM, KVM, and provisioned Linux guest
- ▶ Built in backup and recovery for private cloud storage
- ▶ High security value, high I/O bandwidth, high availability, and greater scale with private cloud.

Other industry OpenStack-based cloud management solutions can also run on LinuxONE, including but not limited to VMware's vRealize Automation product.



Analyze and understand

This chapter outlines some of the points you need to consider before migrating to Linux on LinuxONE. It also provides a description of total cost of ownership (TCO) and helps analyze which workloads would make good candidates for migration. Additionally, it describes the financial benefits of migration.

This chapter includes the following sections:

- ▶ Total cost of ownership analysis
- ▶ Migration planning
- ▶ Choosing workloads to migrate
- ▶ Analysis of how to size workloads for migration
- ▶ Financial benefits of a migration
- ▶ Project definition
- ▶ Planning checklists

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 in most cases determines the magnitude of these costs.

The business case to support a migration to LinuxONE is focused on cost savings provided by server consolidation to the LinuxONE platform and an overall simplification of the distributed environment.

ICU IT in the Netherlands (a Cloud Service Provider) decided to use LinuxONE as their alternative data center architecture for their Linux farms. ICU IT wanted to harness the union of openness, flexibility, and amazing innovation of Linux and open source software with the availability, scalability, security, and robustness of a platform like LinuxONE. With this competitive edge, ICU IT Services has helped businesses slash their IT costs by up to 50 percent and improved their bottom line.

The Met Office is the UK's national weather service, providing weather forecasts for the public, for government, and for businesses in a wide variety of sectors. It creates more than 3,000 tailored forecasts and briefings each day, as well as conducting weather- and climate-related research. The Met Office uses post-processing systems to tailor its weather forecasts for specific clients' needs. The requirement to have the applications run 24 hours a day, 365 days a year, and being able to deliver services day-in and day-out is critical to its brand and its reputation.

Running these systems on a distributed Linux infrastructure was becoming complex and expensive. Therefore, Met Office decided to migrate suitable candidates from its distributed Linux landscape onto IBM LinuxONE.

2.2 Migration planning

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:** After 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 LinuxONE environment, and give special focus to performance.
- ▶ **Education:** The technical staff needs the correct skills to work on a LinuxONE migration and maintain the new environment.
- ▶ **Build Environment:** In this phase, the new infrastructure is readied for migration.
- ▶ **Implementation:** The actual migration. 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.
- ▶ **Post Migration:** 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 a signed acceptance agreement.

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

2.3 Choosing workloads to migrate

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

Consider these variables, among others:

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

Start with a fairly simple application that has a low service level agreement (SLA) and a staff that has the associated skills.

For applications developed within the company, 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. This process will substantially increase the success criteria of the migration effort. Applications that require close proximity to corporate data stored on IBM LinuxONE are also ideal candidates, as are applications that have high I/O rates because I/O workloads are offloaded from general-purpose processors onto specialized I/O processors.

IBM LinuxONE has a powerful processor with a clock speed of 4.3 GHz (IBM LinuxONE Rockhopper) or 5 GHz (IBM LinuxONE Emperor). Because IBM LinuxONE is designed to concurrently run disparate workloads, remember that some workloads that required dedicated physical processors designed to run at high sustained CPU utilization rates might not be optimal candidates for migration to a virtualized Linux environment. This is because workloads that require dedicated processors do 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.

Chapter 5, “Migration analysis” on page 55, provides an in-depth analysis of the process of determining the most appropriate applications to migrate to an IBM LinuxONE environment.

2.4 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?”

The answer to this question is a function of several factors:

- ▶ Processor speed (or speeds) of the servers to be consolidated
- ▶ Average of CPU utilization of these servers
- ▶ Workload characteristics of applications to be consolidated
- ▶ Amount of disk space

Although this process might set limits on the upper boundaries for virtualization, the efficiency of the target platform and platform hypervisor might reduce consolidation ratios. In practice, service levels are often the determining factor.

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

One evaluation tool that IBM offers to the customers is the IBM Rehosting Applications from Competitive Environments (RACE) tool. RACE is a tool that helps IBM to understand the customer environment. To arrange a RACE study, contact your IBM representative.

The following are the inputs for the RACE tool:

- ▶ Distributed server details:
 - Vendor, model, CPU speed, and memory capacity
 - Average peak CPU utilization
 - Workload type (that is, database management system, Java, I/O bound, compute bound, and so on)
 - Consider the following costs:
 - Software license and maintenance costs
 - Hardware purchase and maintenance costs
 - Staff costs

The following are the outputs from the tool:

- ▶ Number of General Purposes Processors (IBM Integrated Facility for Linux (IFLs)) required to support the distributed workload
- ▶ Amount of memory required
- ▶ TCO analysis of the various configuration options (based on cost inputs in the model)

IBM offers another tool to help Chief Information Officers (CIOs) in determining the IBM LinuxONE resources that are required to consolidate distributed workloads. It is a self-help web tool named *IBM Smarter Computing Workload Simulator* that gives a fast and easy way to view areas of potential savings and efficiency through the lens of IBM Smarter Computing systems and technologies.

More details are available at the IBM Smarter Computing Workload Simulator URL:

<http://www.ibm.com/common/sc/simulator/>

2.5 Financial benefits of a migration

In addition, IBM LinuxONE provides costs reduction for the following reasons:

- ▶ Reduced risk of downtime because of the redundancy of the hardware and virtualization features like single system image (specific to z/VM Clustering) and Live Relocation.
- ▶ Save software licensing: Databases, operational systems, application server, and management software in a current distributed server farm can be licensed more cost effectively by using the powerful IBM LinuxONE processors.

- ▶ Save energy and be green: When you have hundreds or thousands of servers that are consolidated in a single box, the energy and cooling costs can be reduced by 75% in comparison to distributed environment.
- ▶ Save costs of on-going support: The complexity of maintenance of the environment is decreased because you have many virtual servers in a single box.

The cost savings arise because LinuxONE 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, significant savings can be achieved by consolidating multiple distributed servers to an LinuxONE processor. Figure 2-1 shows an example company that has 45 virtual servers and uses only 14 licenses.

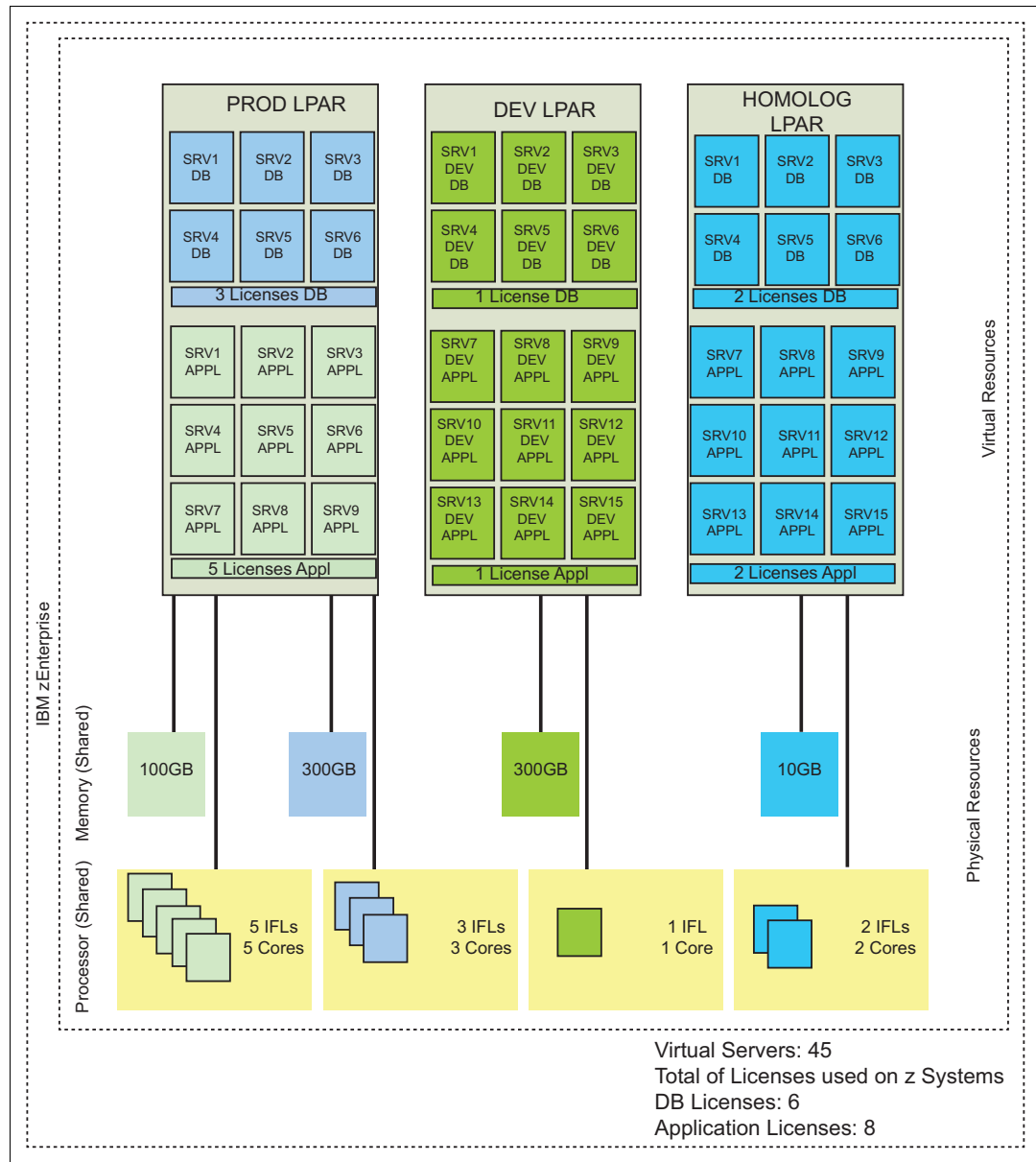


Figure 2-1 Example company saving licenses

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

2.6 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 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.

2.7 Planning checklists

Planning checklists 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 checklist can vary somewhat from project to project or between organizations, the foundation of an effective planning checklist is similar to the generic checklists in this chapter. The checklists shown in this chapter are created specifically for LinuxONE.

2.7.1 Product and tools checklist

The software product and tools checklist template that is shown in Table 2-1 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 LinuxONE operating environment.

Table 2-1 Software products and tools checklist

SOFTWARE PRODUCTS AND TOOLS CHECKLIST				
Name	Version	Vendor/Source web site	License type	LinuxONE

SOFTWARE PRODUCTS AND TOOLS CHECKLIST				
Name	Version	Vendor/Source web site	License type	LinuxONE

2.7.2 Application implementation checklist

The application implementation checklist delivers one level further into the product checklist, where each product or tool is drilled down to their features level. There are scenarios where the same product would not offer the same features on all platforms. These details should be noted in this checklist, as shown in Table 2-2.

Table 2-2 The application implementation checklist

Application name: Database connectivity: Technical application owner:		
	Source (x86)	Target (LinuxONE)
OS name and version		
Architecture model		
Compiler name and version		
Additional software packages		
Observation		
	Compiler options for performance	
Compilation		
Linking		
System library with version		
Shared library		
For debug		
	Compiler options for build	
Compilation		
Linking		
Shared object creation		

Each product or tool that is listed in the product checklist must be analyzed. All the parameters, dependencies, and optimization options must be taken into account in the source operating environment. The planning team must then 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 filling out the application implementation checklist, verify whether changing parameters or options in the target operating environment has any side effects on the application or other tools that are 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 process would in turn reduce the risks and the migration can be run smoothly.

2.7.3 Application environment checklist

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 need to prepare a planning document that lists the resources that the source application needs to provide and all the services that it is currently providing. Table 2-3 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 anything else 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 2-3 The application environment checklist

Source resource	Source location	Target resource	Target location
Internal FTP	FTP server on source application server		
External FTP	Batch process through central and secure FTP server		
Local print	Local print on local LAN		
Remote print	Vendor product secured by host name		
DNS services	Single or multiple DNS servers		
Firewalls	Firewall location and rules exported files		
Internet connectivity	Router location		
Intranet connectivity	Web server location and ports		

Source resource	Source location	Target resource	Target location
Email services	Mail transfer agent co-located on source application server		
Messaging services	IBM WebSphere® MQ on source server		
Client software	User's desktop User's notebooks Mobile appliance		
File services	Type, location, and security		
Log server	Central server location		
SNMP	Agent and server location		

2.7.4 Training checklist

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 correct time.

2.7.5 Hardware planning checklist

The hardware planning checklist lists the hardware resources that you need to consider during a migration project. In the checklist used in this project, the source environment's hardware resources are examined and you needed to acquire similar or more advanced technology that is available for LinuxONE. Table 2-4 illustrates a sample of the hardware planning checklist that we completed for this book.

Table 2-4 Hardware planning checklist completed as an example

HARDWARE PLANNING CHECKLIST			
SERVERNAME:			
RESOURCE	SOURCE	DESTINATION	OBSERVATION
Number of CPU	4	2	Real to Virtual
System memory (in GB)	8	8	
OS SWAP Memory (in GB)	4	4	
Network connection^a			
Connection description	Gigabit Ethernet	Gigabit Ethernet	
Connection type	Gigabit Ethernet	Vswitch/GbE	
IP address/Netmask	9.12.7.88/28	9.12.7.88/28	
Logical volumes:			
Volume Group OS : 20 GB			
Volume Group DB : 150 GB			
Volume Group WAS: 80 GB			
Volume Group MGM: 20 GB			

HARDWARE PLANNING CHECKLIST			
SERVERNAME:			
VLAN number : Vswitch	2	2 : Vswitch1	
Disk resource^b			
OS file system	/ : 30 : Ext3	/ : 2 :Ext4	Root
Mount point: Size (in GB) : Type		/opt : 3 :Ext4 LV OS	Logical Volume
Mount point: Size (in GB) : Type		/var : 5 :Ext4 LV OS	
Mount point: Size (in GB) : Type		/var : 5 :Ext4 LV OS	
Mount point: Size (in GB) : Type		/tmp : 1 :BRTFS LV OS	
DATA file system			
Mount point: Size (in GB) : Type	/DB : 100 : Ext3	/DB:100:Ext4 LV DB	Logical Volume
Mount point: Size (in GB) : Type	/WAS : 50 : Ext3	/WAS:50:Ext4 LV WAS	
CUSTOM file system			
Mount point: Size (in GB) : Type		/MGM:10:Ext4 LV MGM	Logical Volume
Logical volumes: Volume Group OS : 20 GB Volume Group DB : 150 GB Volume Group WAS: 80 GB Volume Group MGM: 20 GB			

- a. For IBM LinuxONE, a number of available network connections are available: Ethernet/QETH, Open vswitch, HiperSockets, and Direct OSA-Express connection.
- b. Use the Logical Volume Manager (LVM) for the Linux environment because it provides flexibility and reduces the downtime of the environment with online resize of the logical volumes.



Virtualization concepts

Virtualization is a highly prized capability in the modern computing environment. Virtualization on LinuxONE offers industry-leading, large-scale, and 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 LinuxONE with those commonly used by x86 distributed systems. The two have many concepts in common, but other concepts are very different. This brief comparison provides terminology, vocabulary, and diagrams that are helpful when migrating workloads to LinuxONE.

This chapter includes the following sections:

- ▶ The demand for virtualization
- ▶ Typical x86 virtualization
- ▶ LinuxONE virtualization
- ▶ Linux guest
- ▶ Guest mobility
- ▶ KVM hypervisor components
- ▶ z/VM hypervisor components
- ▶ Virtualized resources

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, 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 dedicating more virtual resources, such as CPU, RAM, and disk, to a virtual host while the virtual host 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 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 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 power that is consumed in the data center. Virtualization helps to reduce the amount of electricity consumed, and hence reduces cost. Aspirations of a “green” data center can similarly be met in part by using virtualization.

Virtualization has become popular in recent years, with research suggesting that more than half of all workloads in the data center are virtualized¹. 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 Control Program (CP)-67. This innovation quickly grew to become a defining feature of IBM hardware, including all LinuxONE systems.

3.2 Typical x86 virtualization

Figure 3-1 on page 25 shows a 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.

¹ Nemertes Research, “Data Center Dynamics”, Ted Ritter, 27 September 2011

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, and 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.

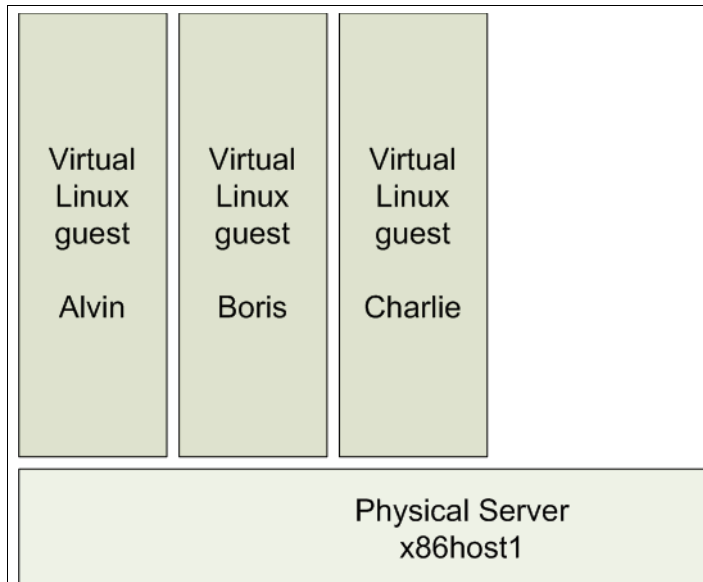


Figure 3-1 Typical x86 virtualization scheme

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 physical server fails, another would be able to take over the running workload with negligible interruption.

3.3 LinuxONE virtualization

Like virtualization systems deployed by using x86 clusters, LinuxONE accomplishes these many virtualization functions. Unlike x86, LinuxONE does so in a consolidated and comprehensive way, with multiple layers of virtualization that provide an extensive list of capabilities for the Linux guest operating system.

A LinuxONE system is carved up into isolated logical partitions (LPARs), with each partition running as an independent system with its own operating environment. This configuration means that it has its own CPUs, RAM, devices such as disks and network connections, and other resources. LinuxONE Emperor can be configured to run as many as 85 partitions, and the LinuxONE Rockhopper can be configured with up to 40 partitions.

With LinuxONE partitions, it is routine to maintain all of the pre-release development of a system contained within one partition, such that anything that goes wrong in this test environment will not adversely affect anything that is running in the production environment. When development is completed, the workloads that have been developed in the test environment can be migrated to the production environment. One partition can be dedicated to development and testing activity, and a separate partition can be dedicated to the production environment.

3.3.1 Process Resource/System Manager hypervisor

The IBM Process Resource/Systems Manager (PR/SM) is a standard feature on all LinuxONE servers, facilitating virtualization of all physical resources of a LinuxONE system into isolated partitions. PR/SM is responsible for the isolation of each partition from each other, and contains powerful tools for the finite management of each.

Although LinuxONE can run dozens of partitions, and thus dozens of Linux instances, greater flexibility to virtualize further can be achieved by running an extra hypervisor within each partition. For more information about these technologies, see 3.3.3, “KVM hypervisor” on page 26, and 3.3.4, “z/VM hypervisor” on page 27.

3.3.2 Dynamic Partition Manager

The IBM Dynamic Partition Manager (DPM) provides a simple mode of operation for a LinuxONE system to easily manage and configure the system and its guests. DPM is not a new hypervisor, but rather a graphical interface that uses the existing PR/SM infrastructure. A system administrator can easily create, modify, and manage the LinuxONE without needing to learn PR/SM and its associated components or commands. The DPM interface allows for dynamic reconfiguration of CPU, memory, and I/O resources. Wizards prompt for specific details about a system as it is being created, and automatically enter those and additional values necessary where they are required. Advanced menus are available for greater control by experienced system administrators.

A system administrator can start the installation of a KVM hypervisor with DPM to simplify the deployment of Linux guests within a newly created partition.

DPM provides some resource monitoring on a per-partition basis, allowing for nearly real-time usage data and historical trends over time.

3.3.3 KVM hypervisor

The Kernel-based Virtual Machine (KVM) hypervisor is a module that can be built into the Linux operating system. KVM enables the virtualization capabilities of a real processor to be used by a user application, which can then set up virtual resources to be used by a guest operating system. On LinuxONE, KVM enhances the capabilities of PR/SM while further protecting and isolating each virtual machine that is established on the host.

Running Linux, extended with the KVM module, within an IBM LinuxONE partition allows multiple instances of the QEMU application, which provides emulation and virtualization to a guest OS. Each QEMU instance runs as a separate process of the host Linux system, thus separating guest instances and protecting each set of virtual resources from each other, and from the host system. QEMU communicates with the KVM interface to establish a guest Linux OS as though it were running on its own private hardware.

For more information about KVM, see 3.6, “KVM hypervisor components” on page 32, and *Getting Started with KVM for IBM z Systems*, SG24-8332.

Using the earlier diagram of a typical x86 virtualization system as a model (Figure 3-1 on page 25), the picture in Figure 3-2 depicts a similar virtualization system as it relates to LinuxONE and KVM.

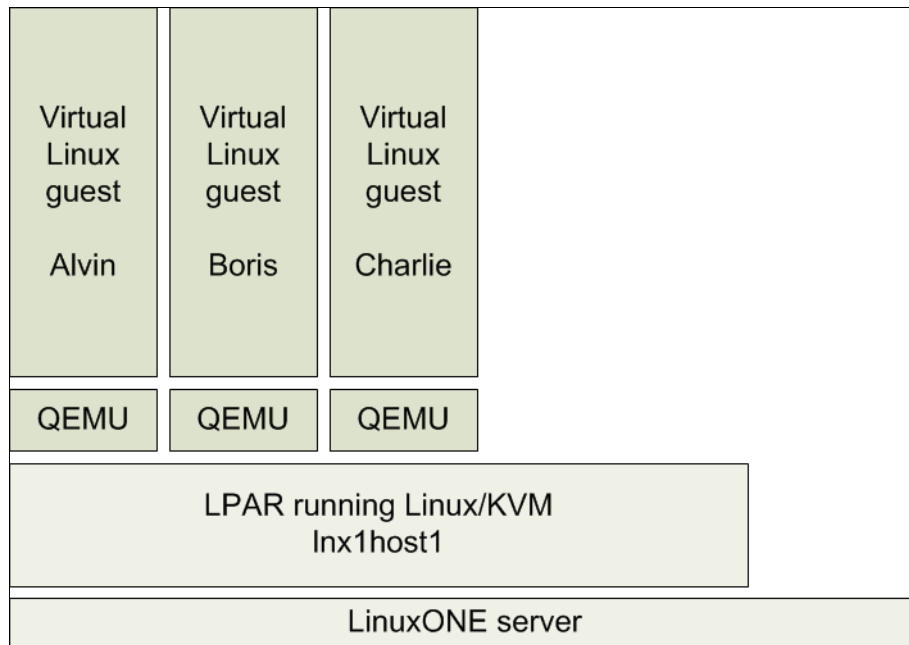


Figure 3-2 Simple KVM with virtual Linux guests

3.3.4 z/VM hypervisor

z/VM is an operating system that facilitates and further enhances the PR/SM hypervisor. A systems administrator might likely know 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. The virtualization capabilities of z/VM provide added isolation, resource sharing, and resource management features that many systems administrators require.

For more information about z/VM, see 3.7, “z/VM hypervisor components” on page 33, and *Introduction to the New Mainframe: z/VM Basics*, SG24-7316.

Using the earlier diagram of a typical x86 virtualization system as a model (Figure 3-1 on page 25), the picture in Figure 3-3 depicts a similar virtualization system as it relates to LinuxONE and z/VM.

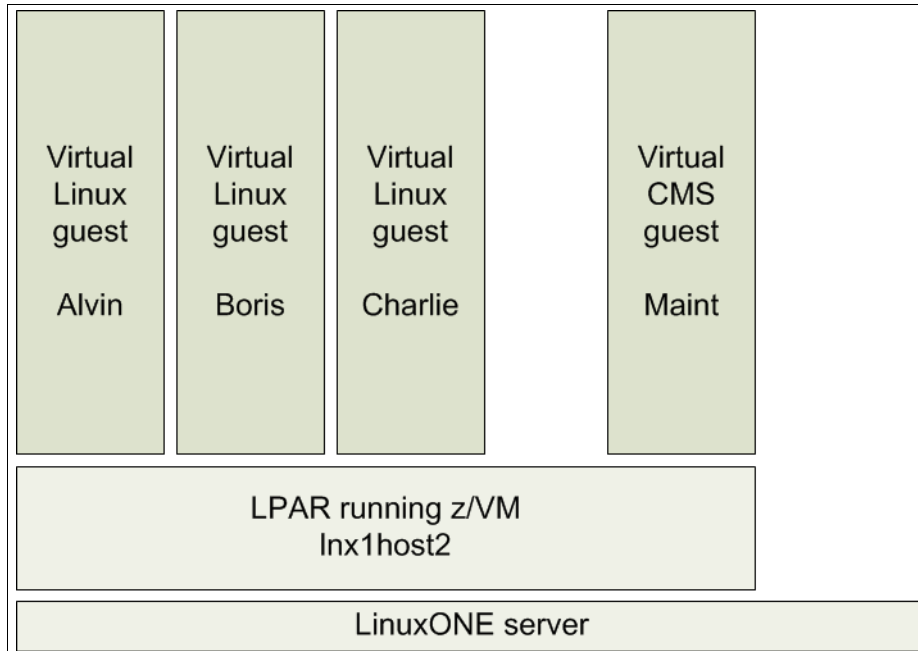


Figure 3-3 Simple z/VM with virtual Linux guests

3.4 Linux guest

Running Linux as a guest under either KVM or z/VM is simple, and Canonical, Red Hat, or SUSE provide Linux distributions that run on IBM LinuxONE 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 LinuxONE hardware.

Figure 3-4 illustrates the work that IBM has contributed to the Linux kernel and the Linux operating system to allow Linux to run on LinuxONE.

Note: All recent Linux distributions that use GNU Linux kernel version 2.6 or later are technically capable of running on LinuxONE. 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 LinuxONE, the distribution must also have binutils, glibc, and other core components built for LinuxONE.

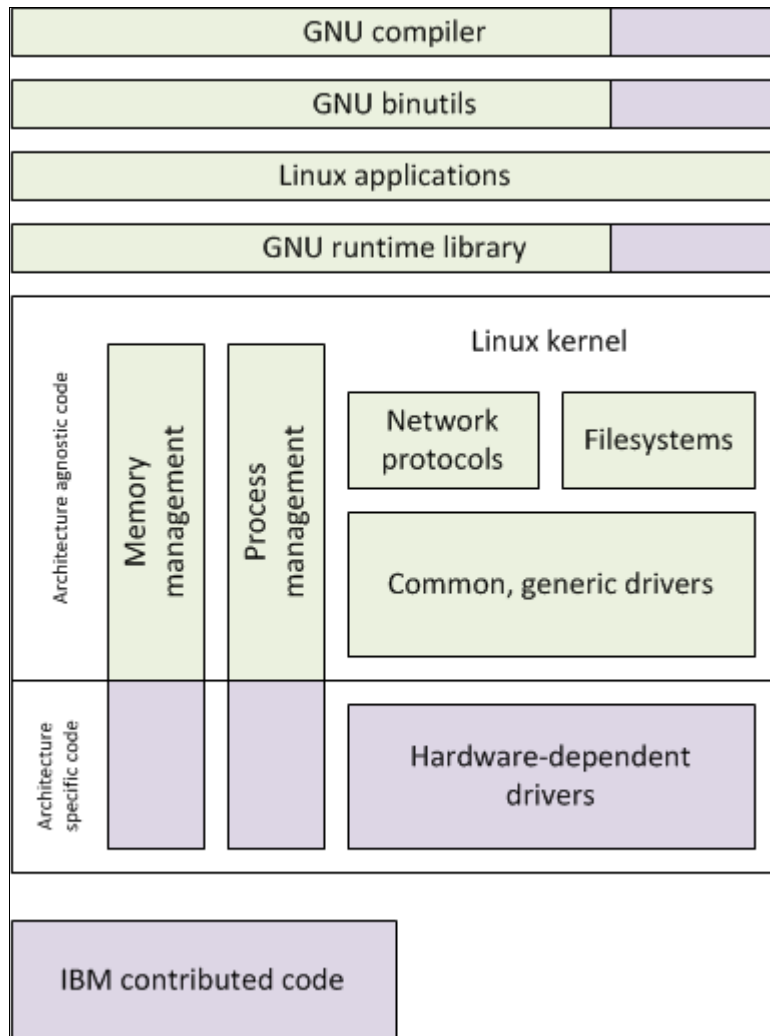


Figure 3-4 Linux kernel and core utilities characteristics on LinuxONE

Running a Linux guest on LinuxONE makes deployment of services faster. You are often able to spin up a running Linux server in a matter of minutes. Linux servers can be built, cloned, and deployed within the LinuxONE 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.

3.5 Guest mobility

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 concern systems administrators. This is a particularly troubling challenge in the data center, where downtime must be minimized and maintenance windows are scarce.

3.5.1 KVM guest migration

The KVM hypervisor allows a migration of guests, either while the guest is running or to pause the system to be restored at a later point. This migration is done by moving the guest state from one QEMU process to another. The guest itself is unaware of the underlying movement. Although it is possible to move a Linux guest to another LinuxONE partition, guest migration is more useful to move the guest to a wholly different LinuxONE system, perhaps because the hardware is being replaced. The migration can be performed between distinct LinuxONE systems, either in an ad hoc fashion or with a high-availability clustering solution that ties multiple systems together as one.

The guest must have access to identical resources on both the source and destination systems to run satisfactorily. An attempt to perform a migration with a mismatch in resources will fail because the guest might not behave correctly upon being started on the destination system.

In addition to being able to move between real servers, the KVM guest migration permits the state of a guest OS to be saved to a file on the host. This process allows the guest OS to be restarted later.

3.5.2 z/VM single system image and live guest relocation

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 common pool disks, network devices, and user data. Ideally the cluster members would be contained on separate systems for optimum safety if a failure were to occur, although running the members on the same system 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 unaware that the underlying infrastructure has moved to a different “physical” environment.

Figure 3-5 depicts a simple representation of an SSI cluster that is composed of two members: Inx1host2 and Inx1host3. Inx1host2 is hosting three Linux guests, whereas Inx1host3 hosts a single Linux guest.

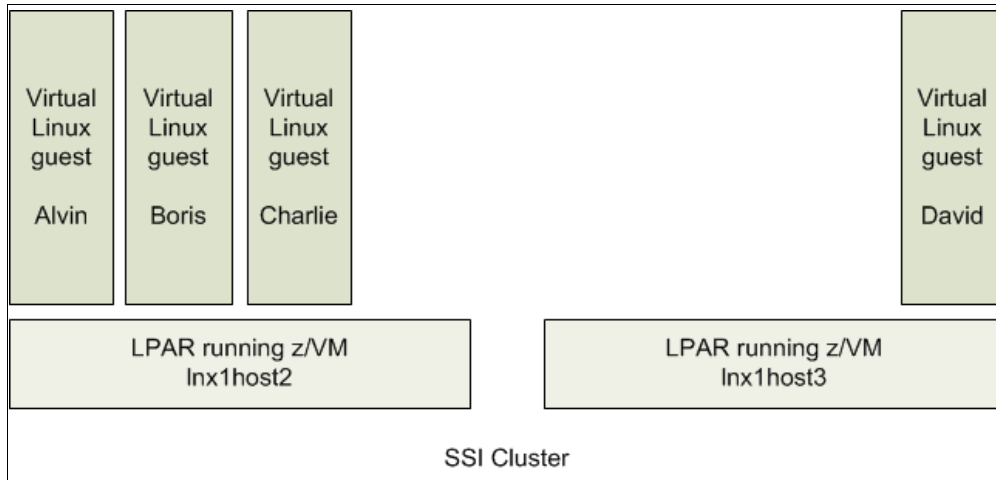


Figure 3-5 Simple representation of SSI cluster before live guest relocation

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 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 can be relocated to a different member of the SSI cluster where network utilization is lower.

Figure 3-6 shows that a Linux guest has been relocated from Inx1host3 to Inx1host2 with no interruption in the services that are running from the Linux guest. Now that there are no guests running on Inx1host3, the host can be rebooted. After rebooting Inx1host3, Linux guests can be relocated back onto Inx1host3.

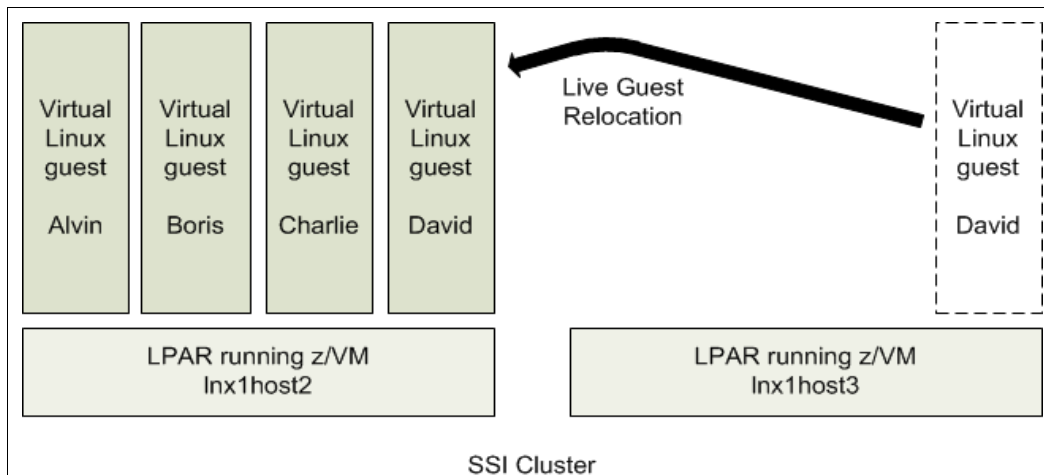


Figure 3-6 A simple representation of live guest relocation of a Linux guest

This is a convenient mechanism of relocating guests with LGR among the various SSI cluster members. Systems administrators need just this kind of flexibility to keep systems up to date while minimizing downtime. This mechanism also gives 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 migrating workloads to LinuxONE all the more compelling.

This section provides a brief overview of SSI and LGR. The following publications describe SSI and LGR in greater detail:

- ▶ *An Introduction to z/VM Single System Image (SSI) and Live Guest Relocation (LGR)*, SG24-8006
- ▶ *The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3*, SG24-8147
- ▶ *Using z/VM v 6.2 Single System Image (SSI) and Live Guest Relocation (LGR)*, SG24-8039

3.6 KVM hypervisor components

“KVM” refers to the entire hypervisor. However, it is just one of many components that make up the entire virtualization solution. This section describes the major components that comprise the KVM hypervisor.

KVM relies on a basic infrastructure known as *virtio* to provide devices to a guest in a high-performance, low-processor requirement manner. These virtio interfaces connect different device types (disk, network, and so on) into a common platform used by both the KVM hypervisor and the guests that are connecting to it before being routed to the respective drivers that communicate with the actual device. For example, a disk would be connected through virtio-blk by the guest before communicating with the Linux block driver on the host.

3.6.1 Linux kernel and KVM module

The KVM module is a cross-platform virtualization technology that extends the Linux kernel into an enterprise-class hypervisor by using the virtualization capabilities of a real processor. In addition, several other modules and packages are included to provide the connectivity and resources necessary for a guest to be run under the host Linux system that runs KVM. This configuration includes, but is not limited to, the KVM and virtio modules that provide an interface to interact with hardware while separating access from individual virtual machines. Further, it provides all packages necessary to provide a useful environment for the system administrator, such as management tools.

The KVM module manages the sharing of all the virtual resources (processors, memory, and so on) onto the real resources, among the different virtual machines that are running in parallel.

3.6.2 QEMU

The QEMU application connects to the KVM interface of the host kernel and provides both emulation and virtualization of resources that are presented to a guest OS run within the QEMU process. The invocation of the QEMU process specifies all the resources that the guest has access to, such as processors, memory, disks, and network devices. An interface

exists that allows certain resources to be added or removed from a running process, for example to add a disk to that guest OS.

3.6.3 The libvirt API

The libvirt application programming interface (API) is a collection of open source software that enables more human-friendly interfaces to the QEMU processes. The API can be used to handle things such as starting/stopping a guest, adding/removing resources, or migrating a running guest to another system. Libvirt includes a set of command-line interfaces (the `virsh` command) for managing both KVM itself and the virtual machines that run within the LinuxONE partition. Libvirt also provides the interface for resource management tools as described in “Resource management” on page 33.

3.6.4 Resource management

Many tools that can use the libvirt interfaces that are described above to manage KVM and its virtual machines. This section describes a few common ones for LinuxONE.

Kimchi and a plug-in named Ginger are open source tools for managing virtual machines by using a web browser. Kimchi is installed on a KVM host, and presents an HTML5 web application to connected clients. This tool allows common configuration changes such as storage and network devices, basic system information and statistics, and simple KVM services such as shutdown and restart.

OpenStack is a collection of services that manage storage, networking, and compute resources for cloud computing throughout the KVM lifecycle.

Virtual Machine Manager (`virt-manager`) is a graphical user interface for Linux that allows a user to manage multiple KVM hosts from a single location. It provides KVM lifecycle management, and some performance metrics of recent virtual resource usage of each virtual machine.

3.7 z/VM hypervisor components

Two primary components of z/VM help PR/SM in managing the virtualization environments. These components are command line operating environments that give the system administrator control over the hypervisor. An additional component allows a graphical interface that can be used for easier administration over the hypervisor and its guests. This section describes and explains these components.

3.7.1 Control program

The control program (CP) provides a user (in this example, the Linux operating system) with a complete environment of a virtual machine 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. There is an entry in the z/VM directory for each virtual machine that can be started.

Each entry contains information about the virtual resources that are required by the guest operating system, and details of the relative priority of the virtual machine. This information is used by CP to determine which virtual machine is to be dispatched. Communication to the Linux system can be either through the Linux virtual machine console (which must be a 327-type terminal emulator), or more commonly by using an SSH client terminal.

Note: If an administrator logs off the Linux virtual console by using the conventional LOGOFF CP command, the virtual machine powers off and terminates all running work. The administrator must use the DISCONNECT command (not the LOGOFF command) to ensure that this problem does not occur.

3.7.2 Conversational Monitor System

The Conversational Monitor System (CMS) is an 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 virtual machine user profile entries in the z/VM environment. CMS is the operating system for many service machines such as TCP/IP, print services, directory maintenance, accounting, and error recording.

For more information about z/VM, see *Introduction to the New Mainframe: z/VM Basics*, SG24-7316.

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

3.7.3 IBM Wave

IBM Wave for z/VM is an intuitive virtualization management software product that provides management, administration, provisioning, and enables automation of Linux virtual servers in a z/VM environment.

To reduce the complexity of z/VM management, IBM Wave for z/VM is a perfect 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 virtual machines, 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 virtual machines to other partitions
- ▶ Display, monitor, and manage z/VM hypervisor resources, such as paging

More information about IBM Wave can be found in *IBM Wave for z/VM Installation, Implementation, and Exploitation*, SG24-8192.

3.8 Virtualized resources

A key feature of LinuxONE is how resource utilization is optimized and maximized. In the current environment of distributed computing, the memory, the CPU, or the disk is underutilized most of the time that the server is running. However, it is necessary to have the capacity available when the server is under peak load. With LinuxONE, a considerable amount of “overcommitting” is possible, such that memory, 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. Overcommit is powerful for virtualized guests because typically not every guest needs all of its allocated resources at the same time.

3.8.1 Virtualized CPU

LinuxONE is equipped with dozens of processor cores, which reflect directly on the performance of the Linux guest running in a partition. The number of virtual CPUs allocated to a single Linux guest should not exceed the number of logical CPUs allocated to a LinuxONE partition. For example, if the partition has four cores, do not allocate five virtual CPUs to a single Linux guest system. If a situation occurs where the Linux guest uses 100% of the CPUs, that will adversely affect the entire partition and all Linux guests that are running within it.

However, in a partition with four cores, you can assign three virtual CPUs to a LinuxA guest and two virtual CPUs to a LinuxB guest, and another two virtual CPUs to a LinuxC guest. All requests for CPU cycles are managed by the hypervisor. Generally, maintain a total of four active virtual CPUs to one logical CPU in a partition.

3.8.2 Virtualized memory

System memory is a resource that is shared across all LinuxONE guests. Each virtual guest is assigned a defined amount of virtual memory during logon.

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. Also, be aware of the amount of real memory (storage) that is allocated to the LinuxONE partition.

Both KVM and z/VM allow you to overcommit memory, but keep the overcommitment ratio of the total amount of virtual memory likely to be active to 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 38.

Memory management features

LinuxONE and its hypervisors have memory management features that you can use to reduce the amount of memory that is required by virtual guests:

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

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

CMM

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

CMMA

CMMA enables a Linux guest to share the page status of all 4 KB pages of guest memory with the KVM or z/VM hypervisor. Linux does this sharing by marking the status of each page, which allows the hypervisor to preferentially steal unused and volatile pages and thus reduce paging.

NSS

NSS is a z/VM feature that 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 is in storage accessible to all virtual machines. This feature decreases storage requirements and simplifies maintenance.

DCSS

DCSS is a z/VM feature that allows virtual machines to share reentrant code for applications, such as Oracle, which also reduces overall storage requirements. Figure 3-7 illustrates how both NSS and DCSS work. Linux guests use a single copy of the application in real memory. The NSS copy of Linux is also shared by all virtual guests.

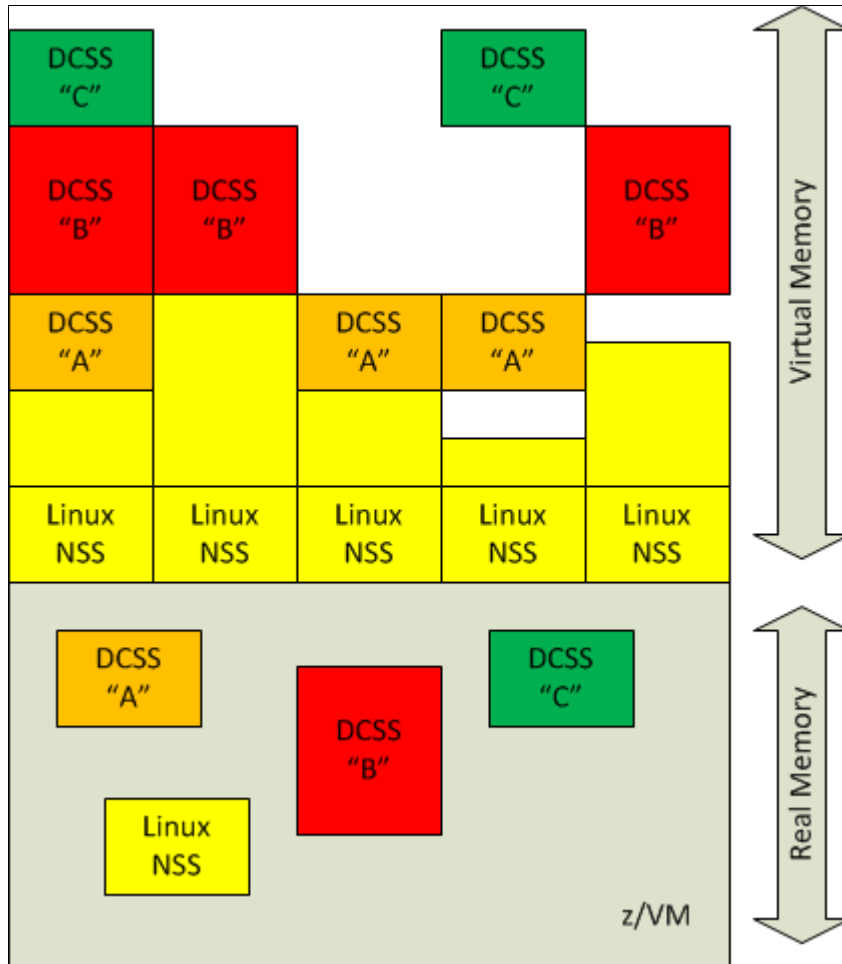


Figure 3-7 DCSS and NSS shared by multiple Linux guests on z/VM

For more information about setting up a Discontiguous Saved Segment and its use with the Execute-In-Place (XIP) file system, see *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 uses all the extra available memory allocated to it as a file system cache. Although this feature is useful on a stand-alone system (where that memory would otherwise go unused), in a virtualized environment this causes the memory resource to be consumed in the partition. Therefore, it is important to assign only the memory needed for the running applications when they are at peak load.

Linux swap can be thought of as an overflow when an application cannot get enough memory resource. Thus, paging is an indication that the application needs to be analyzed to understand whether more memory is needed. Simply adding more memory to a virtual machine might not be the solution to the problem.

Swap device consideration

Understand that the concept of “swapping” is different today than when it was invented, back when large amounts of RAM were very expensive. Modern operating system memory technology is more focused on paging than swapping. As suggested before, commit a specific amount of virtual memory to each Linux guest to accommodate no more than its intended workload, and to fine-tune this amount of memory precisely so that paging does not normally occur. This principle might not be realistic, but it is a something 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 that are allocated in z/VM memory. They become a fast swap device for Linux. Swapping to a VDisk in memory is far more efficient than swapping to regular disk, 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 that 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.

3.8.3 Virtualized disk

Many x86 computing environments have disk storage maintained in storage area networks (SANs) and other similar, external storage arrays. LinuxONE 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 migrating the application workload to LinuxONE. Whether maintaining the data on a SAN or migrating the data to the LinuxONE 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 eases and simplifies the migration effort.

With LinuxONE, SAN device support is expanded to include Small Computer System Interface (SCSI), connected to LinuxONE through Fibre Channel. In the x86 distributed world, the term Fibre Channel is often abbreviated as FC. To avoid confusion with FC in IBM terminology, referring to FICON® channel devices, LinuxONE uses the phrase Fibre Channel Protocol (FCP) to refer to the connection to a SAN.

Typically, a SAN with Fibre Channel consists of independent and redundant fabrics, which provide connectivity between processor and peripheral devices. The Fibre Channel adapters each have their own unique worldwide name (WWN) which is put into zones within the fabric.

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 separate zones, despite sharing physical resources on a server.

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. Although this process does not cause errors, it 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 effect on other zones.

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.

KVM provides SCSI support by using the Linux `zfcp` and `scsi` drivers, and can pass a full SCSI LUN to a Linux guest through the virtio infrastructure. z/VM provides SCSI support by using its own drivers, and can pass either a full SCSI LUN, or a small piece of one as a minidisk, to a Linux guest. However, it is more common for z/VM to pass the FCP devices to a Linux guest, and allow the Linux to perform the SCSI configuration.

For a more detailed description of disk storage, see 5.2, “Storage analysis” on page 66.

3.8.4 Virtualized network

The physical network in LinuxONE consists of devices known as Open Systems Adapters (OSAs). An IBM LinuxONE provides up to 96 OSA-Express5S ports for external network communications, handling up to 640 TCP/IP stacks simultaneously. The OSA supports both copper and fiber Ethernet connections at speeds of up to 10 Gbps.

OSA devices can be virtualized through a virtual switch device to many Linux guests. It is available to KVM guests by using the Open vSwitch interface, and to z/VM guests by using a VSWITCH controller. Each Linux guest connects by using a virtual device that is controlled by the `qeth` module to a virtual switch system in a LinuxONE partition.

HiperSockets provide high-speed interconnectivity among guests that run on IBM LinuxONE, without any special physical device configuration or cabling. The guests communicate with one another internally by using the in-memory capabilities of the PR/SM hypervisor. However, HiperSockets are not intended to be used for sophisticated networking and should not be used for external traffic.

Both OSA-Express and HiperSockets use the queued direct I/O (QDIO) mechanism to transfer data. This mechanism improves the response time by using system memory queues to manage the data queue and transfer between a hypervisor and the network device. Various examples are available in 5.1, “Network analysis” on page 56.

For more information about network in Linux, see *Advanced Networking Concepts Applied Using Linux on IBM System z*, SG24-7995.



Part 2

Migration

After you have decided to migrate from the x86 platform to LinuxONE, this part of the book guides you with an overview of the migration process and assists you in migration planning.

The following chapters describe the key components of a migration analysis and walk you through an example hands-on migration. Planning checklists and worksheets are provided in this part to assist you in your own planning and hands-on migration.

This part includes the following chapters:

- ▶ Chapter 4, “Migration process” on page 43
- ▶ Chapter 5, “Migration analysis” on page 55
- ▶ Chapter 6, “Hands-on migration” on page 121
- ▶ Chapter 7, “Post migration considerations” on page 129



Migration process

In the field of information technology, the term *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 about the approaches that are 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, but the titles that you use should match the functions described in this book.

Additionally, this chapter describes the process for a migration project from identifying the stakeholders, assembling them, and identifying success criteria through to verifying both the migration itself and its success.

This chapter includes the following sections:

- ▶ Stakeholder definitions
- ▶ Identify the stakeholders
- ▶ Assembling the stakeholders
- ▶ Migration methodology

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 ensure 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 just when they are needed to perform tasks for you in the execution stages of migration project.

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 might have a staff member reporting on technical issues, including migration proposals that must be evaluated by the technology stakeholders. Conversely, proposals for migration might 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 13.

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 get achieved post migration project.

- ▶ Users

These stakeholders are the end customers. They use the application or consume the services that are provided by the application, and perform testing to ensure that the application is working at least at 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 must 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 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 CIO. In some companies, the highest level of IT management is 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 is responsible for 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 are managers or supervisors of mainframe system administrators and system programmers. Managers at this level have various types of influence on the migration project. Some projects are originated and championed by these stakeholders. They usually have a high level of technical competence and understanding of the technologies that are 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.

- ▶ Infrastructure administrator, hardware administrator

Infrastructure administrators are responsible for defining and configuring hardware and the virtualization layer in LinuxONE. LinuxONE offers several virtualization technologies: PR/SM, z/VM, and KVM for IBM z Systems.

Unique to LinuxONE, PR/SM is a Type-1 hypervisor that runs directly on bare metal, allowing you to create multiple logical partitions (LPARs) on the same physical server. These stakeholders can easily define and configure hardware definitions by using the IBM Dynamic Partition Manager tool. Using IBM Dynamic Partition Manager, infrastructure administrators can configure and define LPARs and their resources like processor, memory, storage, and network. In addition, the various cloud management capabilities provided by both z/VM and KVM are OpenStack-enabled, which makes the infrastructure administrators' duties more efficient.

- ▶ Linux, UNIX, and Windows administrators

Linux administrators might help installing Linux, 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 are the same as on other platforms, such as Linux on x86.

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

Other services, such as DNS, mail servers, and security, will be running 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 are 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.

For more information about IBM LinuxONE networking, see 5.1, “Network analysis” on page 56. The network concepts and tools outside of LinuxONE is the same for these stakeholders.

- ▶ Database administrators (DBAs)

The tasks that are 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, ensure 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 that are developed in-house require porting and testing on the target Linux system. The effort that is 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. See 5.3, “Application analysis” on page 76 for more information about the issues that need to be considered.

- ▶ Operators

The operators monitor the application, the operating system, and physical environment by checking the monitor consoles, logs, and 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 depends 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 whether 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 LinuxONE, and there are often open source alternatives.

► Contractors

Specialists can be called on to assist with transient needs. They can 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 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 LinuxONE security, see 5.6, “Security analysis” on page 95.

4.2 Identify the stakeholders

The first phase of the migration involves identifying the stakeholders, as defined in 4.1, “Stakeholder definitions” on page 44. In turn, the stakeholders identify the business and operational requirements that affect the migration process. All stakeholders within the company must be consulted to ensure that their requirements are factored into the migration planning.

Identify the following stakeholders, as defined in 4.1, “Stakeholder definitions” on page 44:

- 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 helps to uncover whether extra administrator, manager, or user skill enhancements are needed. The participants are also the people to whom status and milestone results are reported. Some of these people might have never met, and a cohesive, efficient, and successful project requires personal relationships.

To make sure that all interests are taken into account, 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.

4.3.1 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 reject or oppose the project. 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. This perception might cause staff to think that 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 goals. Even the smallest of migrations should be able to do this process, and it will help guide your planning.

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

Often, the migration to Linux is accompanied by other objectives. For example, some customers upgrade their database at the same time to get the latest features and performance enhancements, and to obtain support that works well with the latest distributions of Linux. As with any project, the scope must be well defined to prevent project overrun. However, 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 LinuxONE 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.

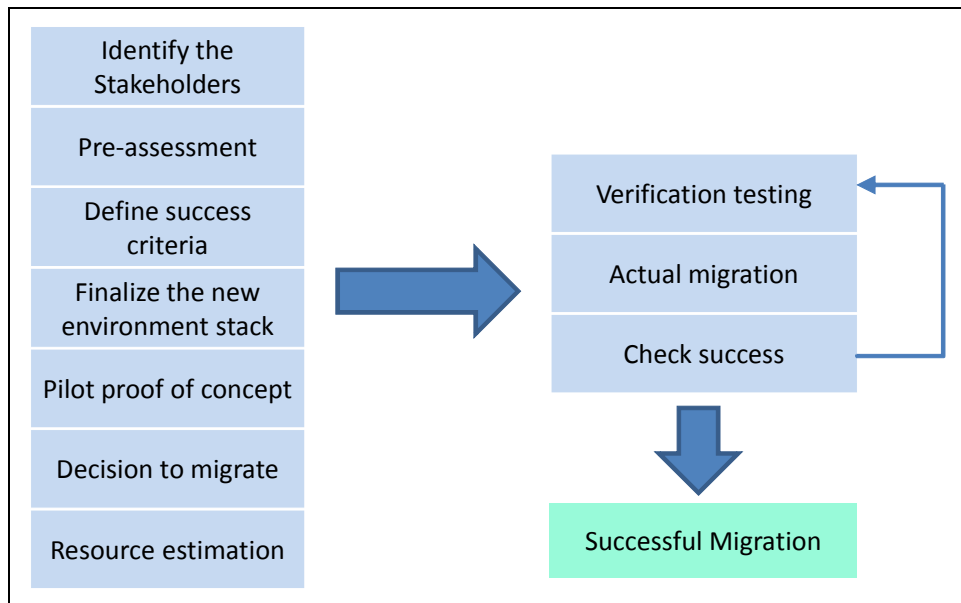


Figure 4-1 Typical migration approach

Identifying the stakeholders is described in 4.2, “Identify the stakeholders” on page 47 and 4.3, “Assembling the stakeholders” on page 48. This section describes 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 that run on the current servers are assessed to determine whether they are available and certified to run on LinuxONE. In addition, an evaluation of the risks that are related to migration is performed. This process helps to identify major risk areas at the earliest stage.

Additionally, perform a careful analysis of present and anticipated business needs and weigh the results against the pros and cons inherent in each option of migration. The outcome of this phase is a recommended migration approach, and a high-level risk assessment and analysis report that identifies 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 might mean, for example, passing a percentage of system tests on the LinuxONE 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 porting cycle must be communicated to all stakeholders and approved before they replace the existing criteria.

4.4.3 Finalize 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.

Generally, 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 are missing, you need to list other parameters available in the tool that would provide the same functions or feature. For examples of forms that can be used to help document your software and hardware requirements, see 2.7, “Planning checklists” on page 18.

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

Custom-built applications

If custom-built applications are written in one or more programming languages, several tools might need to be validated on the target environment. These tools can include compilers, the source code management system, the build environment, and 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 LinuxONE platform provide the expected functionality after the migration (for example, static linking, library compatibilities, and other techniques). The effort that is 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 whether the vendor provides compatible versions that support the distribution and version of the target LinuxONE. Many ISV applications have other third-party dependencies. Be sure to map out all ISV dependencies, including middleware. Most leading middleware and open source products are available on LinuxONE.

Note: There are many open source alternatives for many applications and services for LinuxONE.

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 that are 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 might require extra funding and can 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 and a plan that details the resources, time, and costs that are required to migrate to LinuxONE.

During this phase, 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 include all key requirements (such as resource needs) and business commitments (such as service level agreements) for each stakeholder in the plan.

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 ensuring 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 when performing your resource estimation:

- ▶ Resources

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

- ▶ Education
Identify skills-related requirements and determine whether the staff has adequate Linux education. Decide whether there are special skills that are needed for the areas specific to 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
Determine 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 LinuxONE 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 needed, modifications are made to the application source code and build environment. The new application binary files 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 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?

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

4.4.9 Check against success criteria

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

If the success criteria are not achieved, the migration implementation must be reviewed. After the review is 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.

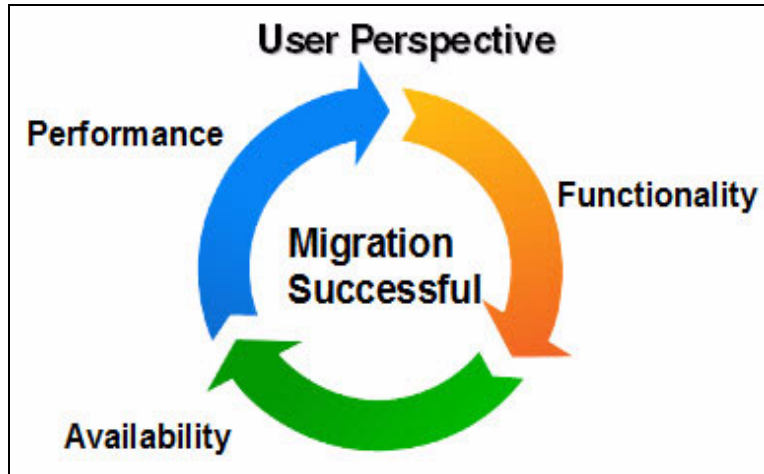


Figure 4-2 Criteria of success from the user perspective



Migration analysis

This chapter helps you to understand new features found on Linux on IBM LinuxONE and provides a technical direction for your migration. Each section addresses a different part of your infrastructure, and uses scenarios to show how the migration affects the environment.

This chapter includes following sections:

- ▶ Network analysis
- ▶ Storage analysis
- ▶ Application analysis
- ▶ Database analysis
- ▶ Backup analysis
- ▶ Security analysis
- ▶ Operational analysis
- ▶ Disaster recovery and availability analysis
- ▶ Virtualized Environment to LinuxONE Cloud Migration

5.1 Network analysis

This section provides information about network migration configuration issues, explains how the virtual network can be configured, and the facilities that are available on LinuxONE and its hypervisors. The following terms and components are used throughout LinuxONE:

- ▶ Open Systems Adapter

The Open Systems Adapter (OSA) serves the same function as an Ethernet card on x86, by acting as the hardware network controller and providing connectivity to clients on local area networks (LANs) or wide area networks (WANs). It can be directly attached on Linux, but will typically be attached to virtual switches. You can find more technical information about OSA cards on *IBM z13 Technical Guide*, SG24-8251.

- ▶ 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. How to configure it is described in *Advanced Networking Concepts Applied Using Linux on IBM System z*, SG24-7995.

- ▶ HiperSockets

HiperSockets is a LinuxONE microcode emulation of a Logical Link Control Layer of an OSA interface, and provides near zero latency at memory speed communications between servers running in different LPARs, but not external connections. If an external connection is required, a HiperSockets bridge must be implemented by using a virtual switch, or a Linux guest must be set up as a router.

HiperSockets provide a fast connection between LPARs. This direct connection without involving real hardware is an important factor to simplify setups with many Linux systems. Some benefits are explained in *Set up Linux on IBM System z for Production*, SG24-8137.

- ▶ 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 LinuxONE, a VSWITCH provides direct attachment of Linux guests to the local physical network segment. The VSWITCH allows IP network architects and network administrators to treat Linux guests as a regular server in the network.

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). MTU 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 server would be.

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 single untagged VLAN on an interface.

5.1.1 Network facilities available on LinuxONE and KVM

The switched network inside a KVM hypervisor is commonly implemented with Open vSwitch (OVS) devices, which provide a robust, multilayer, open source virtual switch. OVS supports standard management interfaces and protocols, and can bond multiple OSA devices together for redundancy. OVS devices can communicate between virtual machines, or between a VM and an external network hosted by KVM.

Another networking device, MacVTap, virtualizes bridge networking and is supported on KVM. However, OVS bridging is generally preferred due to its richer features and a more granular control over devices.

VLAN and VLAN tagging are supported by both OVS and MacVTap devices.

5.1.2 Network facilities available on LinuxONE and z/VM

The switched network inside a z/VM hypervisor commonly is implemented with a VSWITCH, managed by the VSWITCH virtual machine. When running the VSWITCH as Layer 2, it behaves similar to a real switch just between virtual machines.

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

The VSWITCH infrastructure provides two basic configuration options. One configures user-based access, and 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.

5.1.3 Network migration overview

Consider several different levels of network migration because LinuxONE allows for complete virtual network systems. Among other features, you can create multiple virtual switches in the same partition, and the create virtual LANs (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.

This section covers some common scenarios and how they look on LinuxONE.

Single network scenario

One of the most common scenarios is the migration of several distributed machines from the same physical subnet to a single LinuxONE partition attached to the same network segment. Figure 5-1 shows an example that involves a single distributed network.

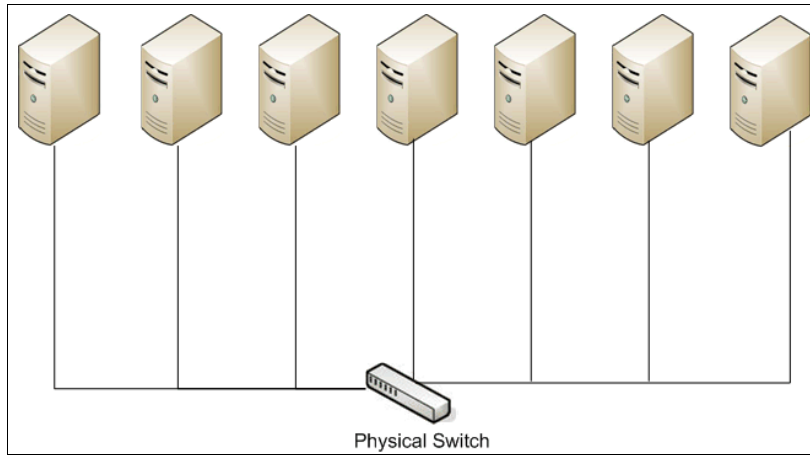


Figure 5-1 Single distributed network

Within this scenario, all physical machines can be migrated to a single LinuxONE machine running Linux and sharing a virtual switch that is attached to an OSA card. The OSA card is then connected to the physical network. Figure 5-2 illustrates this type of configuration.

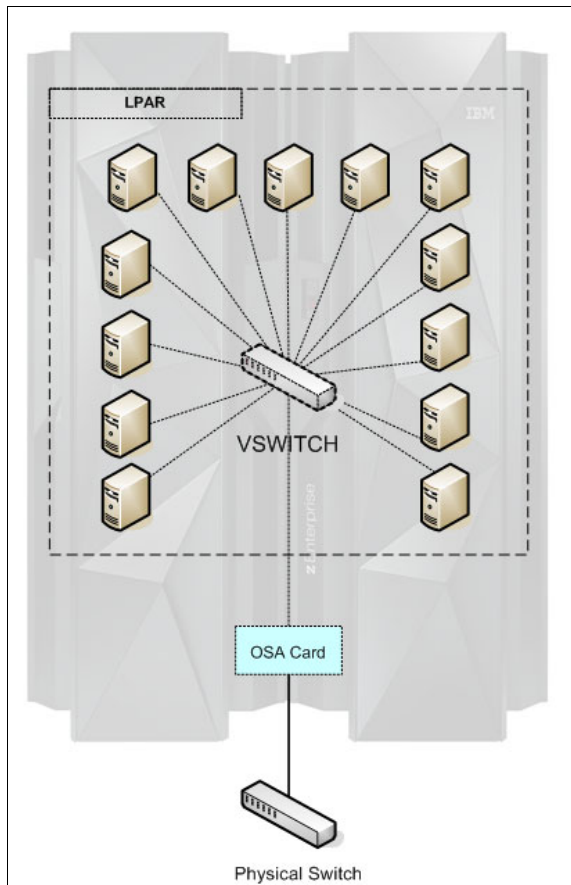


Figure 5-2 Single virtualized network

To increase the availability of each Linux guest, the preferred solution is to configure two or three OSA cards that are attached to different physical switches in the network. This configuration provides a network failover capability, as illustrated in Figure 5-3.

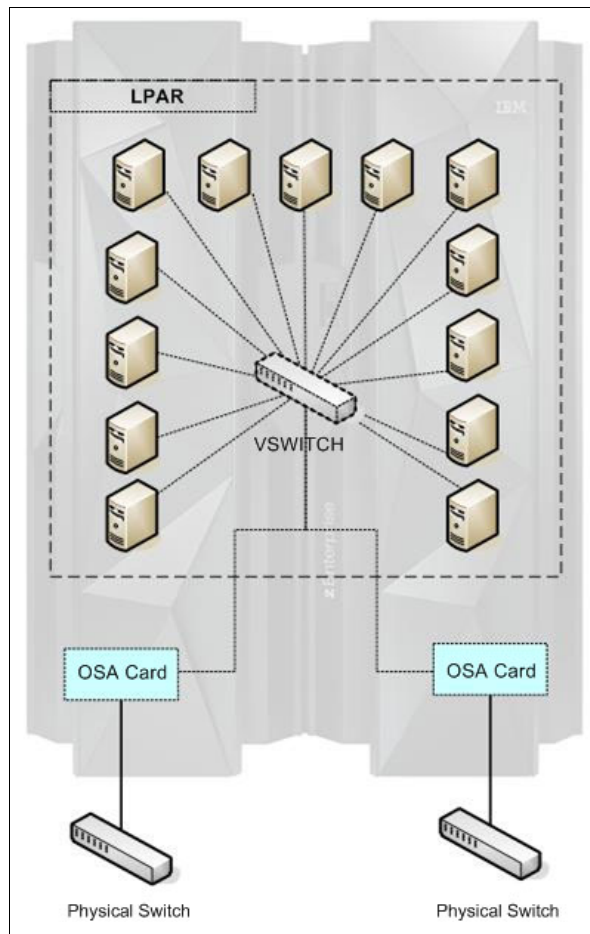


Figure 5-3 Single virtualized network with failover solution

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 LinuxONE LAN segment.

In a multiple partition scenario where a single network segment is used, the preferred solution is to share the OSA card between partitions. Each partition'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 partitions. This configuration is illustrated in Figure 5-4.

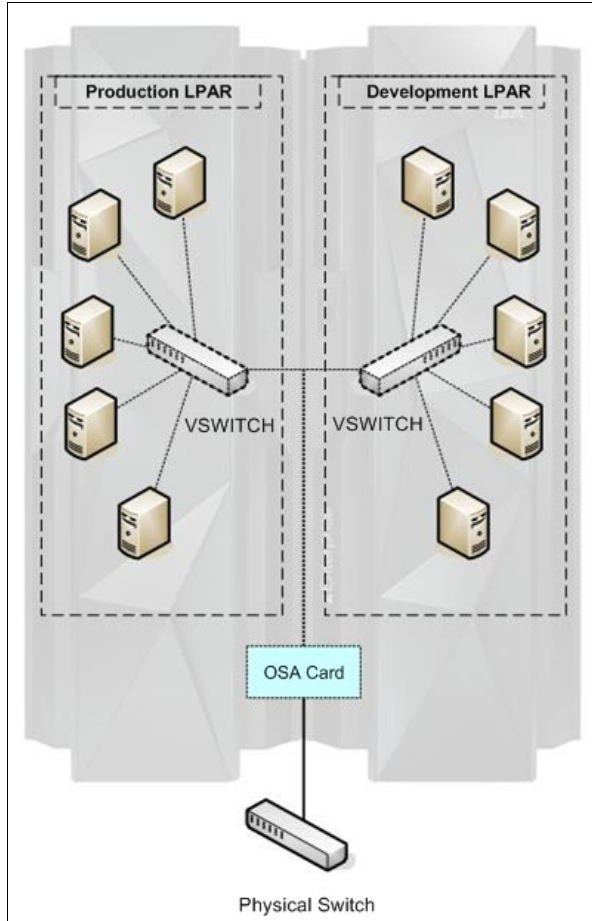


Figure 5-4 Single virtualized network with multiple LPARs

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

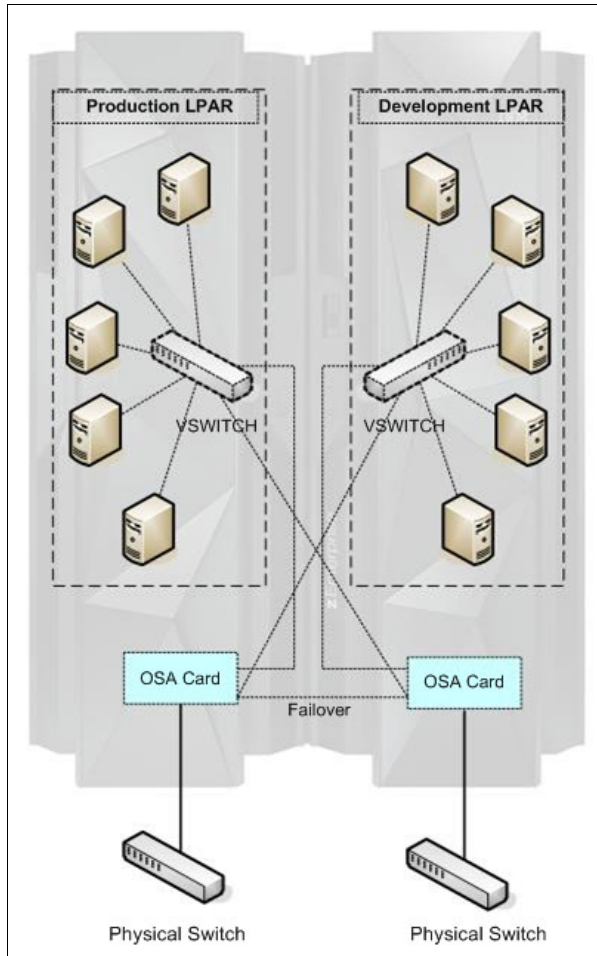


Figure 5-5 Single virtualized network with multiple LPARs and failover

Multiple network scenario

Several types of network solutions require multiple network segments. Some of these solutions 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 LinuxONE and share a physical LinuxONE server. Analyze the DMZ and a secure network scenario. Figure 5-6 shows a DMZ network where the Web Application Server is placed, and a secure network where the database server is located.

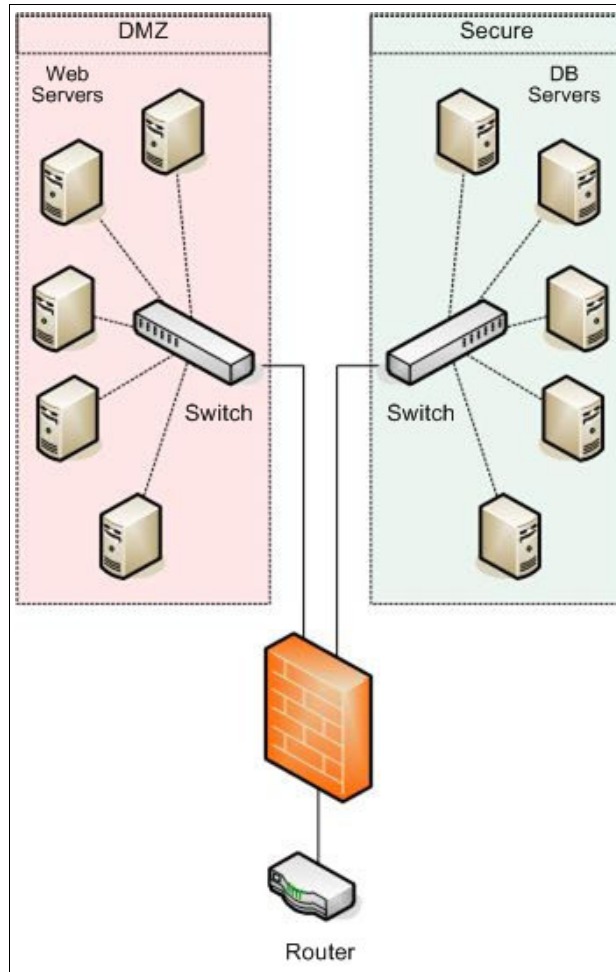


Figure 5-6 Multiple distributed network scenario: DMZ segmented network

You can set up the same scenario on LinuxONE. 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 LinuxONE. Otherwise, you can use some network facilities available on LinuxONE and its hypervisors.

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 are connected to two VSWITCHs in two different network segments. Figure 5-7 shows a network by using a Linux guest as a firewall.

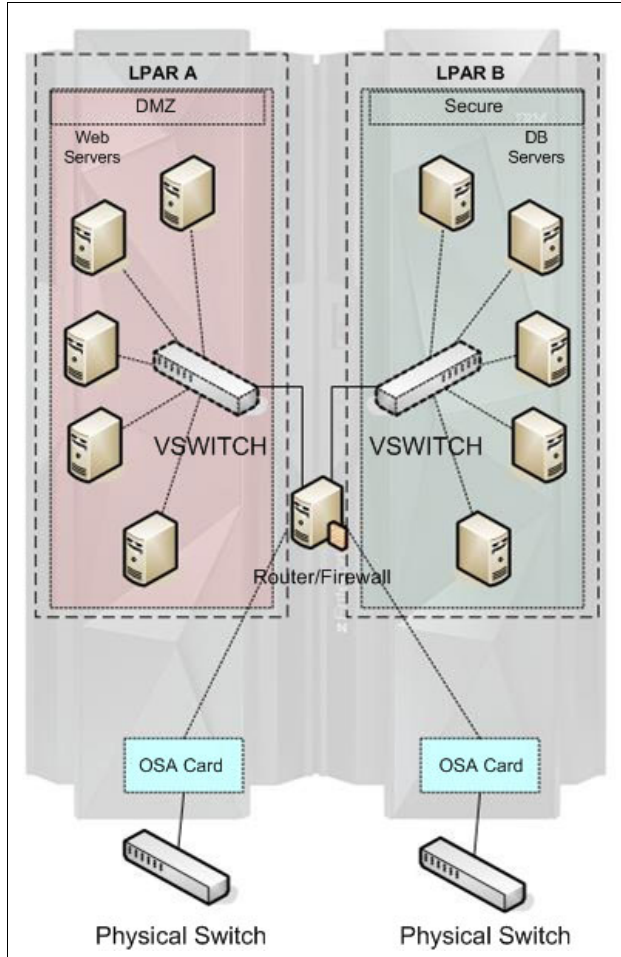


Figure 5-7 Multiple virtualized network scenario: DMZ and secure network

You might have noticed in Figure 5-7 on page 63 that the configuration does not share the OSA cards. It is possible to have the OSA card shared between multiple partitions on the same physical LinuxONE server. To create this solution, generally use an external firewall to manage the network filters. Figure 5-8 illustrates the solution that is described as a network segmented partition.

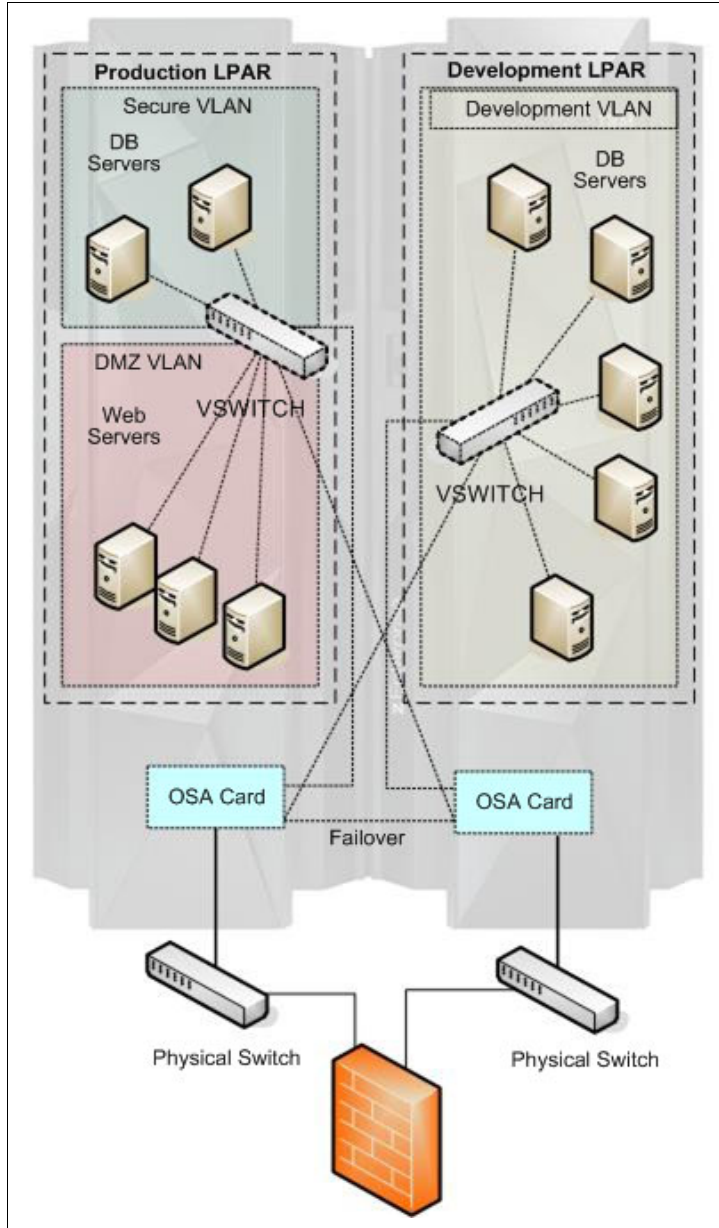


Figure 5-8 Multiple virtualized network scenario with failover: DMZ and secure network

You can isolate the entire secure network from the physical network segment by using multiple partitions. The communication between the partitions is managed by HiperSockets devices. Figure 5-9 illustrates an example.

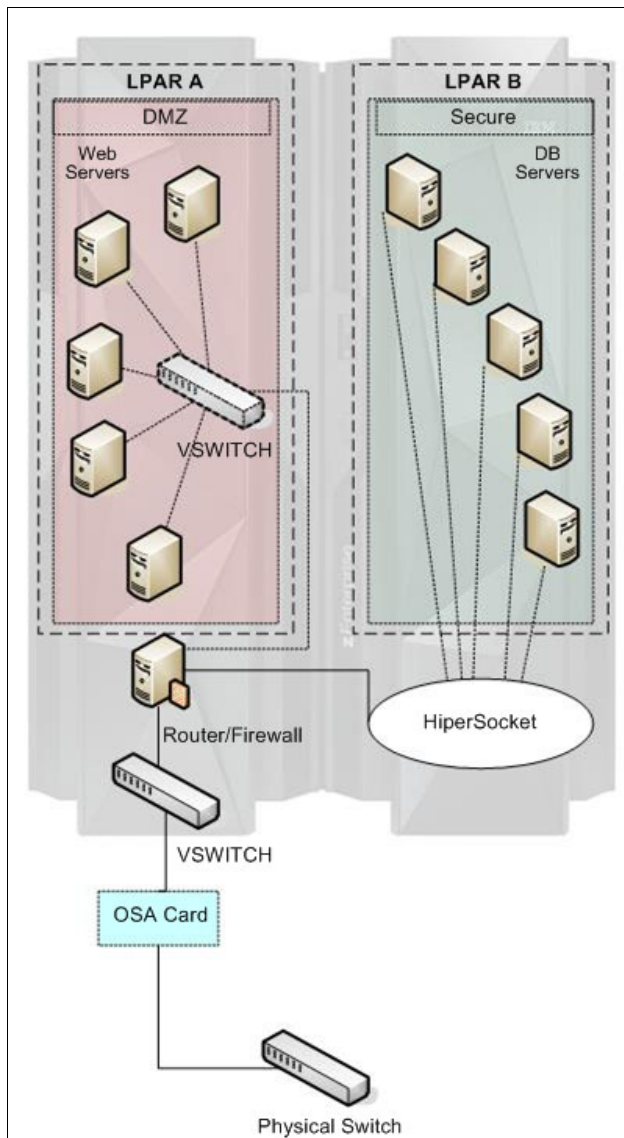


Figure 5-9 Multiple virtualized network scenario with multiples LPARs

Note: Although the use of HiperSockets for this scenario is possible, it might not be the best solution. If one of the LPARs is CPU-constrained, using HiperSockets could cause a delay of network traffic. For more information about HiperSockets, see *Set up Linux on IBM System z for Production*, SG24-8137.

VLAN segmented network

The use of the VLAN tag on virtual switches is supported. The VLAN configuration helps with the segmentation of network packages, bringing security and organization to the environment. It facilitates the administration of the network by grouping the guests with common requirements regardless of their physical location. The VSWITCH, like a physical switch, provides 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 5-8 on page 64, the entire LinuxONE environment is virtualized and all configurations are made per virtual machine, which increases the security, reduces the complexity, and simplifies the environment.

5.1.4 Helpful steps for a network migration

The Linux administrators and network administrators should work together to engineer the best solution for your environment. Complete the following basic steps:

1. Determine the new IP addresses for the new servers. Each 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 servers.
3. Configure the VSWITCH with the listed VLAN IDs.
4. Configure the Linux servers by using the designated IP addresses.

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

1. Migrate the applications (for more information, see 5.3, “Application analysis” on page 76) and files from the source server to the target server.
2. Shut down the source server.
3. Change the Linux server’s host name.
4. Change the DNS registered name to the new Linux IP address.

If the application running is an IP-based application, you can change the IP address of the target Linux server to the source IP address.

5.2 Storage analysis

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

5.2.1 Data migration

Two models of data migration are discussed in this section:

- ▶ *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.

These migration models are covered in more detail in the following subsections.

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

Online data migration

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

To perform an online migration, complete these steps:

1. Install and configure the target Linux guest. See 5.2.2, “LinuxONE: pre-installation considerations” on page 70 for more details.
2. Install the middleware application on the Linux guest.
3. Copy the application data to the target Linux guest.

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 amount of time during the migration process.

4. Include the Linux guest in a cluster as a cluster node.
5. Monitor the Linux guest to verify that the application is responding to requests correctly.
This step is not a test of the application on LinuxONE. The application must be tested on a development system to ensure that the application is a LinuxONE compatible application (see 5.3, “Application analysis” on page 76 for more details).
6. Shut down the source servers.

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

To avoid such issues, online data migration must always be run 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 functions:

- ▶ 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 when using the Rsync application, schedule service synchronization for an entire week before the outage by completing these steps:

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

Run the first synchronization during a time when the use of the server is low. Rsync only copies files that are not locked, which avoids any issues with files in use. However, during this time server response might be slower than normal because of the extra read I/O activity.

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

Only modified files are 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 is able to copy all files to the target servers.

4. Shut down the source servers and start all services on the target Linux 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 run at the target server. See 5.4, “Database analysis” on page 84 for more details.

Migrating storage volumes

When the source servers are Linux x86 and connected to an external storage device using Fibre Channel, and a zFCP device is part of the same storage area network (SAN), you can connect the source Linux volume to the target Linux guest on a LinuxONE server. However, both servers cannot share a volume at the same time.

Storage SAN Volume Controller

One option available to simplify the storage and data migration for Fibre Channel disks that are involved in a migration to LinuxONE 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 5-10 shows where the SAN Volume Controller sits in the SAN. The SAN Volume Controller has visibility to all supported storage on the SAN.

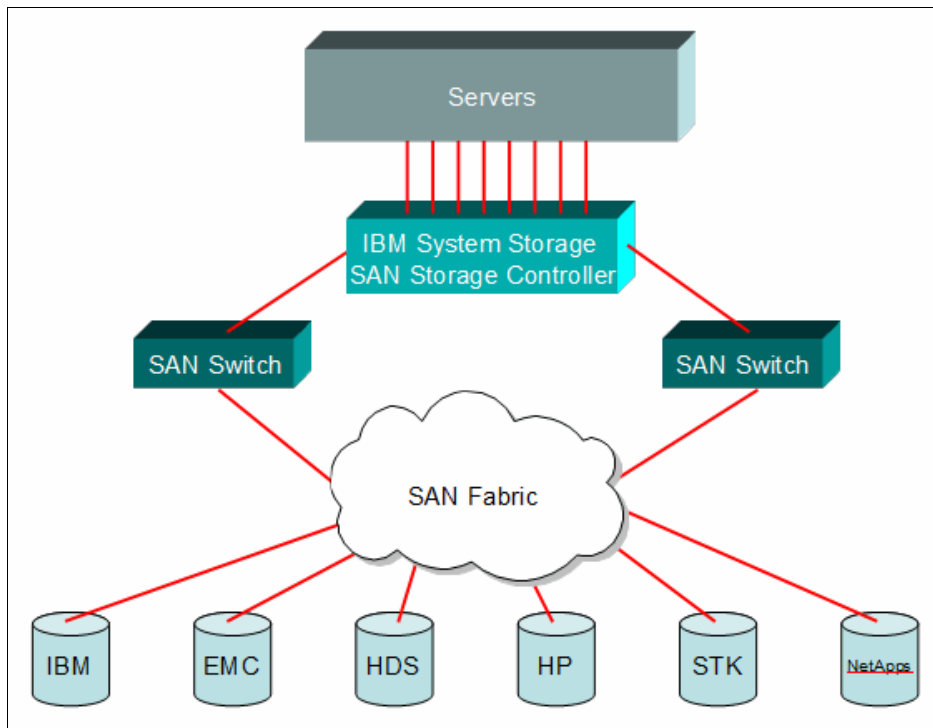


Figure 5-10 SAN Volume Controller

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®
- ▶ Copy data across metropolitan and global distances as needed to create high-availability storage solutions

When migrating Linux systems from x86 to LinuxONE, the SAN Volume Controller allows you to non-disruptively migrate data to LinuxONE. For more information about the IBM System Storage SAN Volume Controller, see the following site:

<http://www.ibm.com/systems/storage/software/virtualization/svc>

Additional information is available in these publications:

- ▶ *Introduction to Storage Area Networks, SG24-5470*
- ▶ *IBM System Storage SAN Volume Controller and Storwize V7000 Best Practices and Performance Guidelines, SG24-7521*
- ▶ *Implementing the IBM System Storage SAN Volume Controller with IBM Spectrum Virtualize V7.6, SG24-7933*
- ▶ *Implementing FlashSystem 840 with SAN Volume Controller, TIPS1137*

Helpful steps for an auxiliary storage migration

The multiple possibilities provided by LinuxONE to store and access files lead to many types of solutions. The solution that you design for the target system dramatically affects the flexibility, efficiency, and performance of the migrated application.

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

1. Create a 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.
4. 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 LinuxONE. The data can be transferred by using an FTP server that is configured on the target server.
 - If not enough space is available 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.
5. 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, you can migrate to a LinuxONE server configured with zFCP adapters to connect directly to the volumes that should be migrated to LinuxONE servers.

5.2.2 LinuxONE: 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 LinuxONE is that only one version of a Linux OS distribution should be installed from scratch. Therefore, design the basic Linux file system to allow the highest possible model of servers and then clone all the other Linux guests in the environment from this source (known as the *golden image*). On IBM Wave for z/VM, this golden image is called a *prototype*. The file system that stores the application data is created after the cloning process depending on the needs of the application that resides on the server. If you want to know how to create an SUSE Linux Enterprise Server 11 or RHEL 6.4 golden image, see *The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3*, SG24-8147.

Logical Volume Manager

All file systems, except the root (/) file system, should be created as Logical Volume Manager (LVM) devices. File systems that are created with an LVM make it possible to expand or reduce the file without a system outage (using SUSE Linux Enterprise Server 10 SP2 or higher, or RHEL 5.0 or higher and LVM2).

The LVM is 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, such as a SCSI device connected over an Fibre Channel Protocol (FCP) device.

- ▶ 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 5-11 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.

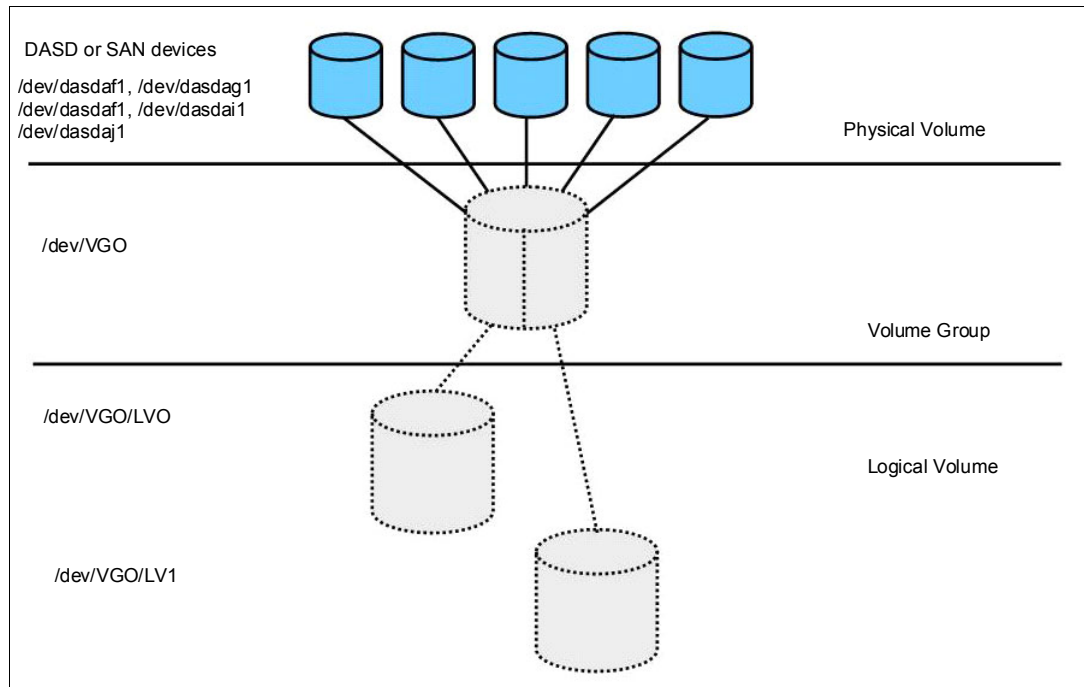


Figure 5-11 LVM example

However, a small performance price must be paid when using LVM. However, the flexibility of LVM often outweighs the cost of the performance loss.

For more information about the LVM setup during the installation, see *The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3*, SG24-8147.

Linux file system

As mentioned previously, design the basic Linux OS file system so that one single image (the golden image or prototype) can be cloned and 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

The following sections describe 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 350 MB is generally enough for the root file system.

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

The /boot file system

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

Important: Like root (/), do not place the /boot file system on an LVM device. The preferred 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 binary files, 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.

Keep the golden image /usr file system size at the minimum to support the basic Linux distribution files. You must be able to increase this file system because after cloning the server, the system administrator might need to increase the file system to install new packages or additional package dependencies.

Create this file system on LVM devices that allow you to dynamically extend or reduce the file system size.

In a shared Linux environment, this file system can be set as read-only because the system simply needs to read the application file into memory. This setup 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 are kept on the server. Also, consider whether the application is designed to write files here, and their sizes and frequencies.

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

Because it is a dynamic file system, place it 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. These files would be deleted every time that the system is rebooted or deleted by the application immediately 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 needs to be increased.

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

The /opt file system

Deploy all third-party applications in the /opt file system. As a preferred practice, further organize the /opt directory by the company or organization that developed the application or software. The next directory level then specifies the software package that is installed. For example, install a DB2 for Linux server at /opt/ibm/db2. Place a WebSphere Application Server in the /opt/ibm/WebSphere directory.

The file system size depends on the size of the software packages that will be installed in it. It is easy to estimate the requirements for a single software package. However, 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 depends on the server function and the number of users who are defined on the server. For example, application production servers do not need a large /home file system because typically 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 need sufficient file system space to create and transfer their files.

Depending on the situation, the /home file system can be a dynamic file system. If it is dynamic, configure it on an LVM device.

Other file systems

An example of additional file systems that might 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 must be created in addition to the file system where the application binary files are 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's 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. Only with this information can you plan to create the devices and file systems for the target system.

There might be file location differences depending on 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

For more recommendations, like volume group and disk naming conventions, see *Set up Linux on IBM System z for Production*, SG24-8137.

Shared file system

The data storage in a LinuxONE 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. However, this configuration might be possible at the hardware level.

In a shared disk environment, remember that the file system changes performed by a guest machine that has the read/write control are only be available to other guests that share the file system after unmount and mount of that 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 5-12, 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/IBMHTTP`, where the web server binary files are installed. For the IBMHTTP service, the log files are redirected to the local `/var/log/httpd` file system. All shared devices are the same device type, and are managed by the z/VM operating system.

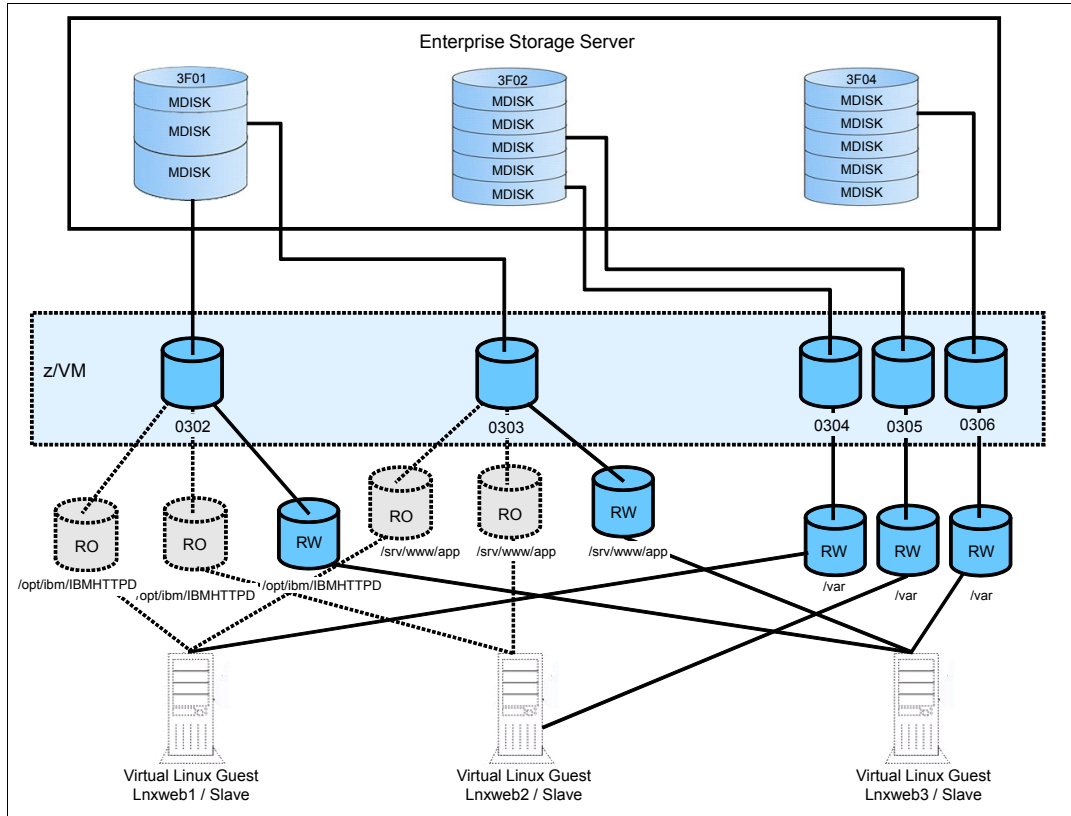


Figure 5-12 Shared devices example

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 then 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 might be lost.

Disk devices

A single Linux guest can talk to multiple disk devices of the same or different device type. This feature is helpful when using large file systems, such as in the case of database servers. A SCSI LUN, accessed by using a zFCP device, provides better response time compared to a similarly configured IBM ECKD™ DASD device, which is common on z Systems platforms. If wanted, you can split up a single disk into partitions with the `fdisk` or `fdasd` Linux utilities, or into minidisks with z/VM.

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.

Software-defined storage

The IBM LinuxONE brings support for IBM Spectrum™ Scale and other software-defined storage technologies. This newer virtualization abstracts the rich features that are found in a single enterprise storage system so that they become available across multiple storage facilities. This feature provides tremendous benefits in the clustering technologies used for High Availability solutions and data replication or backup.

For more information about IBM Spectrum Scale™, see:

<http://www.ibm.com/systems/storage/spectrum/scale>

5.3 Application analysis

This section describes the analysis that you need to perform to identify applications that would be good candidates for migrating to LinuxONE.

This sections deals with the following topics:

- ▶ How to identify the best candidates for a migration to LinuxONE
- ▶ How to select the appropriate application for a LinuxONE proof of concept
- ▶ What you can do if your independent software vendor (ISV) does not support LinuxONE
- ▶ How you can accommodate application interdependencies in a LinuxONE environment
- ▶ How you can redesign your application to take advantage of the strengths of the LinuxONE platform

5.3.1 Why migrate applications

Only perform application migration after thorough planning. You also need 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 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 LinuxONE. In most cases, a migration to LinuxONE 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 is 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.

5.3.2 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 strengths of IBM LinuxONE 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 LinuxONE. This consideration is normally not a problem with homegrown applications, depending on what language they were written in. Also, LinuxONE has a long list of supported open source applications available on the platform.

5.3.3 Selecting an application for migration to LinuxONE

This section lists and describes the basic rules for selecting an application to migrate to LinuxONE.

The following list includes applications that cannot or should not be migrated to LinuxONE, and why they are unsuitable:

- ▶ Applications that are available only on Intel or UNIX platforms.
Requesting an ISV to support their application on LinuxONE is a long process.
- ▶ Servers that have already been virtualized.
In such cases, most of the total cost of ownership (TCO) benefits of virtualization have already been realized and only minor benefits can be achieved. However, if the existing virtualized environment is reaching its limits or the server leases are within 9 to 12 months of expiration, there might be a good business case for moving the applications to LinuxONE because of its higher virtualization capabilities.

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

- ▶ Applications or middleware (database, application servers, and so on) that are supported by a software vendor on multiple platforms, including LinuxONE.
There are no support issues and migration is much simpler.
- ▶ Applications that need close proximity to data on IBM LinuxONE, or that are components of LinuxONE applications.
You can boost the performance and speed of your Linux applications by putting them on the same physical server as their data source.
- ▶ Applications with high I/O or transactional I/O.
Because of its design, LinuxONE excels at handling sustained high I/O rates.
- ▶ Applications with lower sustained CPU peaks and average memory needs.
These are ideal workloads for LinuxONE. The platform has been designed to run multiple workloads at a consistently high CPU and memory utilization.
- ▶ Application development environment for Linux on other platforms.
The virtualized LinuxONE platform provides an ideal environment to test applications before their deployment to Linux on other platforms.

5.3.4 Applications that are best suited for migration

The applications that are described in this section gain the most benefit from the LinuxONE platform 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 use the architectural advantages of the LinuxONE platform.

IBM software

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

LinuxONE offers various solutions, which you can see here:

<http://www.ibm.com/systems/linuxone/solutions/>

Generally, migrating from IBM products on distributed servers to the same IBM products on LinuxONE is a relatively straightforward process. You can see some examples in Chapter 6, “Hands-on migration” on page 121.

DB2

You can use DB2 for Linux, UNIX, and Windows product on LinuxONE. It works seamlessly in the virtualized environment without any additional configuration. In addition, the autonomic features such as self-tuning memory management, and enhanced automatic storage help the database administrator to maintain and tune the DB2 server. To see a migration example from x86, see 6.2, “Migrating DB2 and its data” on page 125.

You can find more information and use cases in *DB2 10 for Linux on System z Using z/VM v6.2, Single System Image Clusters and Live Guest Relocation*, SG24-8036.

Oracle

Oracle database is fully supported on LinuxONE and runs in an efficient manner on this platform, so it is a good candidate for migration to LinuxONE.

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

The Oracle WebLogic Server is also supported on LinuxONE, and allows you to have a complete Oracle Java environment and high availability Oracle database within the same server.

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. To obtain the latest information about which Oracle products are certified for LinuxONE, contact your Oracle representative or refer to the following website:

<http://www.oracle.com/technetwork>

Additional information can be found in:

- ▶ *Experiences with Oracle 11gR2 on Linux on System z*, SG24-8104
- ▶ *Experiences with Oracle Solutions on Linux for IBM System z*, SG24-7634
- ▶ *Experiences with Oracle Database 12c Release 1 on Linux on System z*, SG24-8159

Big data Hadoop solutions on LinuxONE

Big data analytics solutions that are built on the IBM LinuxONE platform are designed to harness the data explosion that is facing the businesses. IBM has taken a leadership role in offering optimized solutions that are ready for immediate deployment. These solutions are built on IBM InfoSphere® software like BigInsights® and Streams. IBM has also taken its leading role in the open source community seriously. IBM has made heavy contributions to projects like Apache Hadoop, enabling continuous development in the fields of analytics and high performance computing. Clients and solution builders who want to innovate on top of a high-performance data analytics platform can take advantage of the flexibility, throughput, and resiliency of IBM LinuxONE Platform, and the immediate price-performance value provided by LinuxONE solutions.

MongoDB solutions on LinuxONE

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 ahead of time and contend with its restrictions. 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.

A LinuxONE server running Node.js and MongoDB can handle over 30 billion web events per day while maintaining 470K read and writes/sec. The popular MEAN stack runs up to 2x faster than on other platforms. IBM LinuxONE allows MongoDB to scale vertically with dynamically allocated resources, instead of horizontally by sharding and replicating the database. LinuxONE and MongoDB provide strong consistency, ensuring that critical data remains consistent, and minimize sharding-related processor usage.

5.3.5 Other software

This section lists some non-IBM software that are good candidates for migrating to LinuxONE.

Infrastructure services

The following infrastructure services are good candidates for LinuxONE:

- ▶ Network infrastructure, FTP, NFS, DNS, and so on, are well served on LinuxONE. These workloads are generally minimal, but are critical to the business. The main benefit of hosting these services on LinuxONE is the availability of the hardware's disaster recovery capabilities.

Additionally, a significant amount of network traffic is generated between data on IBM z/OS® and FTP and NFS servers. When the servers are hosted on the same system as the data and HiperSockets is used, this network traffic is greatly reduced, and the batch processing window for that data can also be reduced.

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

Application development

The following are the benefits that LinuxONE provides for 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.
- ▶ 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 can easily be cloned. You do not 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 another virtual server, the appropriate profile just 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.

For more information about using the Linux environment for application development, see *Linux on IBM eServer zSeries and S/390: Application Development*, SG24-6807.

To obtain an extensive list of applications and solutions across all industries from over 60 ISVs that are certified and supported on LinuxONE, see the following site:

<http://www.ibm.com/systems/linuxone/solutions/>

5.3.6 Selecting an application for a proof of concept

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

1. Talk to other customers who have migrated applications to LinuxONE 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, keep it as simple as possible because a proof of concept is performed to demonstrate that an application can be successfully migrated to a LinuxONE 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. In addition, choose an application that does not require a major rewrite to run on LinuxONE.

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 might have to make a number of code changes.

Applications that are written in C/C++ are also suitable if you have the source code because they must be recompiled for the IBM LinuxONE platform.

After you select an application to migrate, you must also define the end objective or success factor. The minimum objective would be to produce results that are identical to the production version.

5.3.7 Application interdependencies

Not many applications are self-contained. In most cases, an application obtains data from a number of 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. An application migration to LinuxONE provides an opportunity to simplify your application without affecting any interdependencies.

Many distributed applications have grown, in only a few years, from a single server to tens or even hundreds of interconnected servers. These interconnected servers not only add network burden, but also add 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. LinuxONE supports all modern communication methods, so 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.

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.

5.3.8 Successful application migration

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

5.3.9 Special considerations for migrating a Java application

Migrating Java applications from one platform to another is easy compared to the migration effort required for C or C++ applications. Even though Java applications are operating system-independent, implementation and distribution specifics need to be considered:

- ▶ Most of the Java distributions have their own Java virtual machine (JVM) implementations. There will be differences 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 needs to be verified for a similar switch in the target Java environment.
- ▶ Even though Java SE Developer Kits (JDKs) are expected to conform to common Java specifications, each distribution will have 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 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.
- ▶ Ensure that the just-in-time (JIT) compiler is enabled.
- ▶ 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.

5.3.10 Special considerations for migrating C++ applications

When migrating C++ applications, there are a few special considerations to be aware of, as explained in this section.

Architecture-dependent code

Programs residing in directories (on non IBM LinuxONE systems) with names like `/sysdeps` or `/arch` contain architecture-dependent code. You will need to reimplement them for the hardware architecture to port any of these programs to LinuxONE.

Assembler code

Any assembler code would need to be rewritten. Opcodes would have to be changed to s390 opcodes or, if the code uses assembler header files, you would need an appropriate 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 can be downloaded at the following site:

<http://www.gnu.org/software/binutils>

ptrace and return structure

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

Little endian to big endian

LinuxONE is a big endian system, storing 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 have to be reworked to adjust for processing little endian data.

Stack frame layout and linkage specific to LinuxONE

For details about stack frame layout and linkage specific to LinuxONE, refer to `/usr/src/linux/Documentation/Debugging390.txt`. The location of this file varies depending on the distribution. If you are not able to find it, try the kernel documentation:

<https://www.kernel.org/doc/Documentation/s390/Debugging390.txt>

Changes to build scripts

You will need to make appropriate changes or updates to the `Configuration/build/Makefile` scripts or files, and a requirement to add support for the LinuxONE platform.

/proc filesystem

The `proc` filesystem has some differences:

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

Available languages and compilers

Additionally many popular programming languages are available such as Ruby, Perl, and Python.

Shared objects

Linux currently does not support shared objects like mutexes, semaphores, and conditional variables across different processes.

5.3.11 Middleware, libraries, and databases

Any middleware or libraries that are needed must be available for LinuxONE. Supported databases include examples of MySQL, Postgres, Oracle, DB2 UDB, and IBM DB2 Connect™. As described in 5.3.4, “Applications that are best suited for migration” on page 78, a bunch of middleware is available for LinuxONE, like Apache, Tomcat, vsftp, and more. You can acquire it from the package manager or the official website of your Linux distribution.

5.3.12 Helpful steps for an application migration

A successful application migration depends on the combined efforts of the developer team, the network team, the middleware administrator team, and the LinuxONE team. Without the cooperation of all these groups, it is difficult to achieve a successful migration.

You might find helpful these steps helpful during your migration:

1. Perform source application mapping.
Start by analyzing the source application, focusing on its suitability to migrate. Keep in mind the following points:
 - a. Is the source code available to be compiled and deployed on the target server?
 - b. Is there a version of the middleware available for LinuxONE?
 - c. Are there performance reports of actual development tests to compare with after the migration?
2. Design the network solution for the application (see 5.1, “Network analysis” on page 56 for more information).
3. Design the file system for the application and middleware (see 5.2, “Storage analysis” on page 66 for more information).
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.

5.4 Database analysis

This section provides information about the configurations of the database server on LinuxONE. Preferred practices for different database management systems are also presented. And while this topic is presented using offline migration methods, consult your DBMS vendor to investigate the variety of tools available to move data while online.

5.4.1 Before database migration

Database servers are well suited for migration to LinuxONE. However, a migration of the database server also demands detailed planning because technical configuration changes need 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.

5.4.2 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 LinuxONE network (follow steps 1 - 4 as listed in 5.1.4, “Helpful steps for a network migration” on page 66).
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 server.
6. Shut down the source server.
7. Change the Linux server host name and IP address.
8. Perform import procedures at the destination server.
9. Perform the database and applications tests.

5.4.3 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 LinuxONE

A significant factor in the migration of multiple servers to LinuxONE is the distribution of server peak load. Document and compare peak workload information, including how long the workloads take and how much of the server resources are used. Use Table 5-1 to map server workloads when creating the migration configurations.

Table 5-1 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 3.8.1, “Virtualized CPU” on page 35, the CPU and memory constraints in an LPAR are possible and desirable. However, the server should maintain the same peak load for a long 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 causes a CPU constraint because the number of virtual CPUs is four and the number of real CPUs is three.

In this case, the hypervisor handles all the processor requests and the server is still available. However, the performance of the application would probably not be 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 is not affected.

This kind of constraint is acceptable if it happens in intervals of milliseconds to seconds. However, it can become a problem 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 LinuxONE.

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 does not necessarily have four CPUs in the destination virtual server. Preferred practice shows that the actual number is less than that. For more information about this topic, see 5.4.4, “Technical considerations” on page 86.

Migrating a multiple instance server to LinuxONE

Usually on your development environment you have one database server with multiple instances. This configuration generally works well for a development environment, but when you are migrating your production environment, you want to isolate your instances and simplify the database management. For best results from this type of migration, perform a detailed workload analysis. 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 affects only the database server where the instance is running. As a result, only the database server where the instance is running would need to be restarted or investigated.

To minimize the work related to database software fixes and security updates, it is possible to use shared devices for database binary files and libraries. For more information about these topics, see “Shared file system” on page 74.

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

- ▶ 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 5-1 on page 85 to map the instances used by changing the Server name column to “Instance name” and documenting the appropriate information.

With this information, you can configure multiple servers in a partition to respond to all user requests, without degraded performance and with improved database management. This process makes it 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.

5.4.4 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 must be considered at server migration.

CPU

The number of virtual CPU resources in a database server is important. Setting the maximum does not guarantee better performance. The number of CPUs must 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. Therefore, before deciding that the server does not have enough CPU and adding a 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, reorganize the file system.

In the beginning, you will not know how many virtual CPUs your database will need on LinuxONE. Start with a low number of CPUs and increase them 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 large memory area to achieve acceptable performance, but with LinuxONE, allocating more resources is not related to improving performance. Instead, size the machine as needed. One consideration that is 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 size from the source server and then increase or decrease as needed.

Swap memory

In database servers, have a swap area and count it as part of the total usable memory. However, use the swap area only at the peak size to avoid the Linux kernel stopping the database process because of memory constraint.

A LinuxONE preferred practice is to use the z/VM VDisk devices as swap devices. Because swap configured at VDisk devices provides desirable response times, 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 configuration involves 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 might not be the most desirable solution when considering database servers. A configurable kernel parameter called `swappiness` determines whether more or fewer pages of memory are swapped out to disk. The `swappiness` configuration is a whole number from 0 - 100, inclusive, with a default of 60. Earlier versions of Linux suggested setting this value to zero for database workloads. However, changes in newer versions of Linux would take this setting as an indication to almost never swap memory to disk. This configuration leads to a greater risk of out-of-memory conditions for the Linux guest, a clearly undesirable attitude.

Newer Linux distributions such as RHEL 6.4 might suggest a lower, nonzero `swappiness` value, such as in Example 5-1.

Example 5-1 /proc/sys/vm/swappiness

```
at /etc/sysctl.conf file include the line  
vm.swappiness = 10
```

Some other distributions, such as Ubuntu, suggest leaving this value alone and taking the system default. In addition, some databases such as MariaDB have their own suggested nonzero setting. While you monitor your application's performance, some investigation and consultation of your particular environment is needed as it relates to this setting.

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 set your swap size to a safe number that hold this peak and avoids an outage due out of memory issues. 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, monitor swap when the usage peak occurs and increase or decrease it accordingly to improve performance.

Shared memory

Linux systems use the interprocess 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 performs less I/O.

To obtain the best memory allocation, set some Linux kernel parameters depending on what the DBA allocated in the migration. Follow the guidelines in Table 5-2 to avoid issues like memory starvation.

Table 5-2 Guidelines for kernel parameters

Parameter	Description	Guideline
kernel.shmmax	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.	Convert the shmmax value to 4 K (shmmax value x 1024 /4)
kernel.shmni	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 varies depending on your application.
kernel.sem	Four values must be set in this parameter: <ul style="list-style-type: none"> ▶ The number of semaphores ▶ The maximum number of semaphores ▶ The maximum number of semaphores operations within one semop call ▶ The limit on 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 is based on the IBM Knowledge Center website:

http://www.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.kc.doc/welcome.html

You can change it to meet 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 I/O processing capabilities of LinuxONE, the first consideration in design is to spread the I/O workload over as many paths as possible of the storage server.

For more information about how disk device accesses are made and how an external storage system provides its own disk page caching, see 3.8.3, “Virtualized disk” on page 38. If such functionality is not used, the Linux OS spends CPU cycles with disk page caching.

5.4.5 Migrating DB2 and Oracle from x86 to LinuxONE

The following sections provide an overview of the steps that are needed to migrate DB2 and Oracle from x86 to IBM LinuxONE. You can find a full example of migrating your DB2 data in Chapter 6, “Hands-on migration” on page 121.

Migrating DB2 databases across platforms

Although 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.

Four file formats are 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 that is named `employee.ixf` contains uneditable DB2 UDB interchange format data. Import can 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

Before 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 following SQL query to get the exact name of all table spaces. You 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:

```
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, and create the respective tables, indexes, and so on by using the DDL Scripts.

Note: Before importing the dump file, you must first create your table spaces by 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 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 downtime if your database is large. Oracle has developed additional methods to migrate from one hardware platform to another:

- ▶ Transportable table spaces: This technique was introduced in Oracle 8i to allow whole table spaces to be copied between databases in the time it takes to copy the data files.
- ▶ Data Pump export/import: These are high-performance replacements for the original Export and Import utilities.
- ▶ Recover manager (rman): An 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.

5.4.6 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 6 GB of shared memory needs 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 partition 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. Have 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 are needed at both the Linux and database levels.

5.5 Backup analysis

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

5.5.1 Introduction to backup and archival concepts

This section provides a high-level introduction to the basic data and storage management paradigms that are used 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 by using 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 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 under these circumstances:

- ▶ The number of versions stored exceeds the defined limit
- ▶ After a defined time

On any system, several categories of data objects need to be backed up, each with different retention needs. A database table can be backed up frequently at regular intervals, whereas an operating system configuration file would be backed up only when it is changed.

Common backup types

The following are the most common types of 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 that have been created or changed since the last normal or incremental backup. Performed periodically, this backup is generally faster than a normal backup and marks files as having been backed up to assist on the next incremental backup. If you use a combination of normal and incremental backups, you need the last normal backup set and 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 time. Typically, you would use this function to create an extra 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 extra stored object that is normally kept for a specific time, such as vital records.

5.5.2 KVM backup

Because the KVM hypervisor is a module that is loaded into a Linux kernel, the same tools described for a Linux guest in 5.5.2, “KVM backup” on page 92 apply to the Linux running in a LinuxONE partition and acting 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 the Linux guests.

5.5.3 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 Linux volumes here as well. Linux can be backed up while online using FLASHCOPY without issues. For options for backing up data within Linux that offer better flexibility, see 5.5.2, “KVM backup”.

You can see examples of z/VM backups in *Set up Linux on IBM System z for Production*, SG24-8137.

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. It is integrated with Tape Manager for z/VM, can compress data during the backup, and supports encryption exits.

You can find more information on the official website:

<http://www.ibm.com/software/products/en/backupvm>

5.5.4 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 very 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 properly.

Other utilities are available that have customized the use of the command-line tools mentioned above. Amanda, for example, was designed to add a more user-friendly interface for the backup and restore procedures, making backup tasks a little easier to manage. It has both 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 all Linux distributions.

Another handy feature of Linux is represented directly in the capabilities of the file system. File systems such as ZFS and BTRFS are capable of taking 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 themselves, 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. Read more about IBM Spectrum Protect at the following site:

<http://www.ibm.com/software/products/spectrum-protect>

5.5.5 Migrating backed-up and archived data

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

Why migrate archived data?

The following factors can force the migration of archived data, among others:

- ▶ 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, technology changes and your solution becomes less competitive compared to emerging ones.
- ▶ Some older storage technologies have a direct impact on the volume of data that can be stored and the space requirements due to the low MBytes/cm³ and Weight/MByte factors.
- ▶ End of support for your current solution.

5.5.6 General archival migration considerations

Multiple ways of migrating data from the existing operating environment to another operating environment are available:

- ▶ 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 will not be any incompatibility issues with the archived data. This fact also means that the migration will be relatively straightforward because the storage backup and archival manager product are able to access the existing archived data.

Often backup and archival managers have built-in migration tools that migrate the archived data from the source operating environment to the target environment. This is a useful time at which to reorganize the archives and purge unwanted data so that you 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 must be replaced. The hardware can have a highly efficient virtualization server and the software can have new technologies that are either proprietary or open source.

5.5.7 Migrating to new backup software

This section describes the migration approaches that you can employ when changing the target environment's software stack.

Your old software is not compatible with LinuxONE

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

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

Your old software is compatible with LinuxONE

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

5.6 Security analysis

This section provides an overview of the security considerations you need to include in analyzing programs and functions that are going to be part of the migration. Available enterprise-wide authentication options and their possible role in migration is also described. Finally, because SSL/SSH is probably going to be used, the use of the cryptography hardware that is available is covered as well.

A properly secured application, on a properly secured system, is a cornerstone of any computing environment being used today. Thus, migrating an application from one server to another, regardless of the platforms that are involved, is simply a matter of validating that the same policies are available and in place in the new environment. This section covers the security options that can be implemented for an application that is migrated to LinuxONE, if none were already in place. It is entirely possible that these tasks are already implemented, and need validation as being intact after a migration.

This section discusses the following topics:

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

5.6.1 Security migration overview

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 policy is where each individual is only granted the access that they need, no more. You need to establish individuals and their roles, and who is going to be allowed to do what. This process 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 the 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, instead use open, established ones that have been vetted for many years. Hiding information creates more system administration work, and any mistakes could cause a failure to protect against attacks.

System logs and 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 comes into question if a hack is suspected. Therefore, it is important 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 can reasonably be expected to occur. Also, document security plans such as change control, audits, and procedures for break-ins in all domains.

Understanding the hypervisor foundation

The Linux virtual machine (VM) is controlled by a hypervisor, perhaps KVM or z/VM. 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 operating system images to run on the same hardware at the same time. The hypervisor allows for resources to be shared between each VM. It also allows for virtual devices to be created and consumed, like HiperSockets.

LinuxONE and existing security policies

Most organizations have an existing security policy dictating that certain hardware, especially those hosting virtual environments, must not be Internet-facing. With the migration of a distributed environment to LinuxONE, this requirement often raises questions about the role of LinuxONE 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.

Processor Resource/System Manager (PR/SM) has been certified through the Common Criteria at Evaluation Acceptance Level (EAL) 5+. More details about Common Criteria are covered in 1.2, “Reasons to choose LinuxONE” on page 4.

To further ensure the isolation of one partition from another, dedicate the OSAs used to connect to external networks by a hypervisor to the partition in question. These precautions ensure that other guests or partitions cannot share an external-facing network. However, if the security policy states that nothing on a virtualized environment 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 the hypervisor.

Firewalls and existing security policies

In many cases, an organization’s existing security policy identifies specific firewalls that have been approved for use on the corporate network. Most often these are hardware firewall appliances. Although the LinuxONE hypervisors 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.

LinuxONE provides two options. The first is to implement a software firewall on a virtual server within the virtual Linux environment. This configuration has some challenges because the firewall software might not be used in the organization and as such would have to be certified, which might 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 OSA, go through the physical firewall, and then return to the virtual environment through a different OSA. Figure 5-13 illustrates the use of an external firewall.

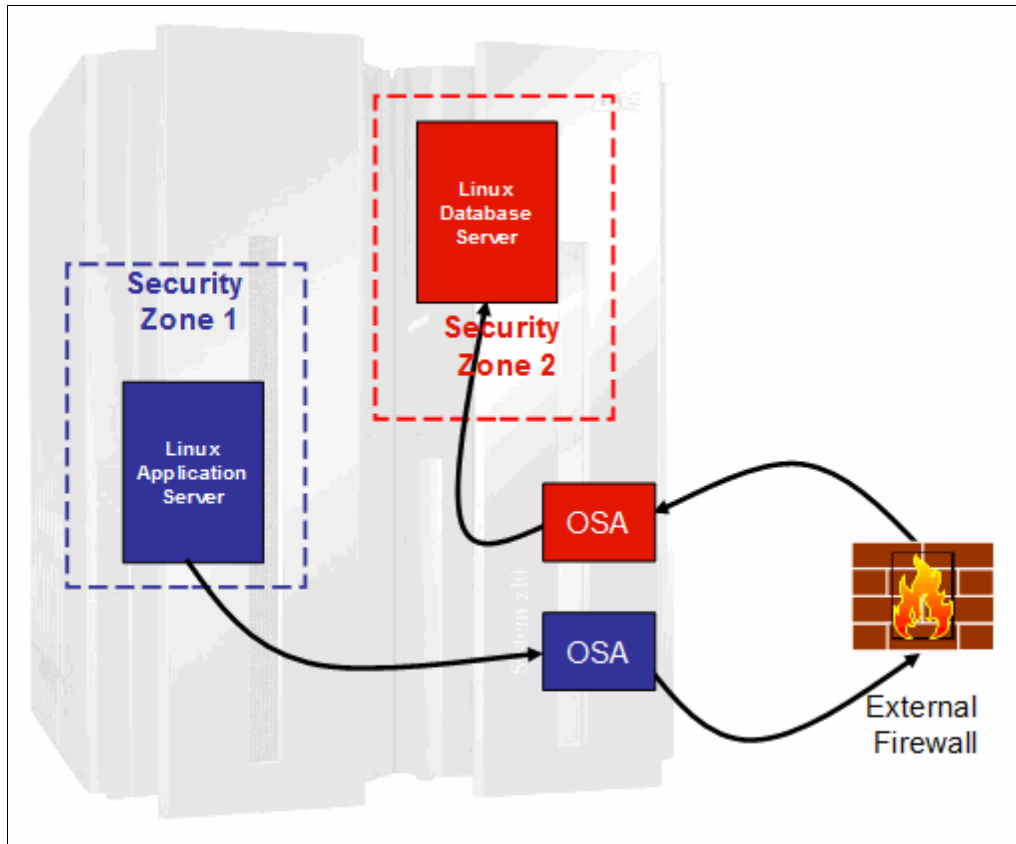


Figure 5-13 Using external firewalls between security zones

In Figure 5-13, the different security zones shown can be in separate partitions or even in the same partition. Customers have reported that using external firewalls has minimal performance impact.

As mentioned, conforming to the existing security policy can simplify a migration. However, the reality is that for applications within the LinuxONE footprint, as shown in Figure 5-13, there might be no requirement for firewalls if all incoming communications to LinuxONE are processed by external firewalls.

Control of hypervisor

Who will own the hypervisor, and what is the protocol for requesting changes or actions? Regardless of whether the KVM or z/VM hypervisor is used, if you control the hypervisor, you need to fully understand it because it is the basis for all the virtual machines on that partition. It must be secure and its access should be highly controlled. Also, document a change request protocol and publish it to all stakeholders.

You also need to plan for hypervisor maintenance, which might require that some or all of the virtual machines be quiesced. Therefore, ensure that a change window is set aside to allow for maintenance, and put a plan in place and set a schedule to allow for security and hypervisor updates and maintenance.

Security references

For more information about hosting Linux guests, and security and networks, see the following IBM publications:

- ▶ *Security on z/VM*, SG24-7471
- ▶ *Introduction to the New Mainframe: z/VM Basics*, SG24-7316
- ▶ *IBM z Systems Connectivity Handbook*, SG24-5444

Hardening the base Linux guest

The term *hardening* is commonly used in server security to mean the process of taking a generic operating system and changing it to only provide what is necessary for the production environment. This configuration provides a baseline for security for the operating system.

During migration, you might be given an already hardened Linux image. In that case, you simply need to know what is allowed and not allowed with the image. However, if a hardened Linux image does not exist, create and maintain one.

Creating a hardened Linux guest

The basics of hardening a Linux on any platform consists of removing all unnecessary applications and services, and then securing the applications and services that remain. Explaining this process is beyond the scope of this book, but for more information, see *Security for Linux on System z*, SG24-7728.

Migrating to a hardened Linux guest

A hardened Linux should have most if not all applications and services removed or disabled. There might be more than one hardened Linux 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 reenabled. If any applications are to be installed and services enabled, you 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: RHEL includes the SELinux security method, SUSE Linux Enterprise Server includes AppArmor for its enhanced security method, and Ubuntu uses AppArmor by default (although SELinux is available). Determine whether those environments are in use or required, and plan accordingly.

Those mechanisms are complex, so invest the time that you need to identify code and applications that have not been ported to work in these environments.

Maintaining a hardened Linux guest

It is necessary to maintain base hardened Linux guest. 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.

5.6.2 Code and application analysis

When migrating an application from x86 to LinuxONE, the vendor might provide a supported package for LinuxONE. The application might only be available as source code (from a vendor, or something developed internally) that needs to be recompiled for the new LinuxONE platform. Either way, it is important to know what the current security methods are, and how they will be used in a virtual machine on LinuxONE. Be sure to poll the stakeholders about the security requirements that must be met.

When moving an application to LinuxONE, consider isolating applications on different virtual machines. If possible, consider isolating application virtual machines from those that will be serving data. The more isolation, the more straightforward the security will be.

If you know that there is a security issue with an application, do not use it. You need to address all security issues before the system is placed in production. If more secure ways to configure an application are available, invest the time to make those changes during migration. For example, you might place a database on a different virtual machine than the application that uses 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.

Code dependencies

Almost all code uses APIs and other libraries to carry out the tasks that it was designed for. Therefore, 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

Generate and review a list of all application dependencies for known security issues. Only fixed or known secure versions should be used. You might be tempted to migrate the application over to the new Linux guest and test to prove that the migration is achievable. However, such testing is invalid if any application or its dependency is on code that has known security issues.

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. Never pass raw input to another application or system request.

Use exceptions to help ensure that input always conforms to the format that is expected, and, if the unexpected occurs, that it can be gracefully handled.

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 run 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.

A security issue might need 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, but instead verify that they are within the bounds of the setting. If they are not, handle the error gracefully.

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 are used by the code, and for what purpose?
- ▶ What type of protocol is used: TCP, UDP, ICMP, and so on?
- ▶ Are special settings used on the port, as in TCP keepalive?
- ▶ How long can a connection tolerate a lack of response?
- ▶ How long is a connection allowed to idle?

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

The syslog-ng daemon will run on Linux. Take time to update the code as needed to send messages to this daemon. At the very least, log all information that deals with security and critical state information. The benefit of implementing syslog functionality is that log maintenance is performed by the system (log rotation and archiving).

Escalations of authority

Apply the “principle of least privilege”, which means that programs only operate with the authority needed to accomplish a goal. 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

Analyze your code to determine any 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

Programs that run as root, the super user, must be carefully examined and assurances given that they are operating as designed. Generally, do not allow any code or program to run with such authority, if you can avoid it. 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 during development, only to discover during testing that certain functions do not work.

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. You need to validate the security assumptions made. This validation determines what you need to include in 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 might be to patch the fault and restore data from the backups (assuming that they have not been compromised).

5.6.3 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 not available to unauthorized users. 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.

Implementing executable system security requires 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 individuals are usually anonymous. Therefore, every effort must be made to mitigate malicious access if you have an internet-facing Linux system. You must be able to identify individuals and their IP addresses so that, if necessary, you can work with the networking team to prevent malicious access while still allowing authorized users to have access.

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 potential security threat needs to be addressed.

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 might try to cover their activities by erasing log files or incriminating evidence.

Making log files immutable

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, and 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 approach in 8.8, “Deploying central log server” on page 172. The logs on the remote log server are not necessarily immutable, but they are not directly writable from a system that has been compromised.

Audit trails encompassing all security domains

Make sure that security audits can always be passed 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, use “authentication” conversations:

- ▶ Users must be assured that they are connecting to the server that 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.

Disabling Telnet access and using Secure Shell (SSH) accomplishes this authentication. Using Secure Sockets Layer (SSL) with web servers also accomplishes this level of security, and is preferred over the default of no SSL.

5.6.4 Data integrity and confidentiality

A benefit of migrating to LinuxONE 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 fits 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
- ▶ Is there 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 a disaster recovery plan is in place

Protecting data at rest from unauthorized access

Protecting access to a database is well understood, but what about protecting raw data on the disk itself? Mobile computers with databases full of accounts or data are sometimes misplaced or stolen. Thus, you need to protect data “at rest” (meaning the files themselves) and ensure that the data is kept in a secure way. Prevent offline copies of a database from being kept on portable devices or drives. Control of data is key. Be sure to communicate the data integrity policy to all individuals who have access, and monitor the audit trail to make sure that the policy is being enforced.

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.

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 is to be used or shared. In addition, a system needs to be in place to enforce the policy. This enforcement is normally done by auditing access logs. If a violation is detected, it needs 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 Payment 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 5-2 to communicate with your system users.

Example 5-2 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 corprate *
*      /'\_/_/^- \   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 5-3.

Example 5-3 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 these items:

- ▶ 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.

5.6.5 Security change management

No system is perfect so there will be changes, however infrequent. Because security fixes are important to keep current, you need 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 remains that way. Outages are expensive for companies, but failing to plan change windows and downtime also causes 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 guest

The advantage of migrating to LinuxONE is that you can clone a VM and test changes before you apply them to the production images. Run through the complete change from start to finish, rather than assuming that it will work.

Record how long it takes to make changes and test worst case scenarios. After testing the change on the clone is complete, report to production stakeholders how long the change will take and how long the worst case will take.

5.6.6 Enterprise authentication options

Migrating to an enterprise system means that user and identification management can be consolidated. This section describes enterprise authentication options and where to find the corresponding information about how to implement them.

A common centralized LDAP server

When migrating applications and code to LinuxONE, you can simplify user administration by storing user information in a Lightweight Directory Access Protocol (LDAP) server. Configuring the Linux guest to authenticate from a centralized LDAP server provides the following benefits:

- ▶ User management is simplified, and users can be managed across the enterprise.
- ▶ Changes made to a user are applied across all images.
- ▶ An offline VM might contain outdated user information. Using LDAP ensures that bringing an old image online does not compromise current security.

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. For more information about this topic, see *Open Your Windows with Samba on Linux*, REDP-3780.

5.6.7 CP Assist for Cryptographic Function

When migrating to LinuxONE, the underlying hardware can accelerate cryptographic mathematics. 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 processor in IBM LinuxONE, or the Crypto-Express card, if it is installed.

The following APIs are supported:

- ▶ OpenCryptoki
- ▶ OpenSSL
- ▶ Global Security Kit

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 use 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 uses OpenCryptoki for hardware encryption.

Using this approach offloads the cycles and allows for more concurrent access to a web server that is using SSL or applications that use one of the supported APIs. To learn about how to configure your system so that your Linux guest takes advantage of the installed hardware, see *The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3*, SG24-8147.

5.7 Operational analysis

The source application comes with a complete support structure. Part of that support structure performs daily operational tasks. Depending upon the application, this support can be 24 hours a day, 7 days a week, and 365 days a year. The application relies on manual and automated intervention to start, stop, monitor, and maintain the services provided by the application.

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 can highlight characteristics of the application that were not clear from the analysis of other application interfaces or from the code itself. The application code might be successfully ported, but it is just as important that the application's operational support structures be migrated successfully as well.

5.7.1 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.

- ▶ Notification of problems

Sometimes automated messages or indicators can be sent that are unfamiliar, and the correct response is unknown. Operators and systems support staff 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 not available 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 process 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 so that they can learn by observation and perform required tasks.

- ▶ Hardware manipulation

Some migrations include hardware consolidation or replacement. Operators 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 tell them 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 LinuxONE.

For example, with the source application and server, an operator might be comfortable with and even required to restart a server by using the power switch. However, on LinuxONE it is a serious error to use a power switch to react to a server or application problem.

If a new hardware vendor is involved in the migration project, the method that the operators must use to notify the vendor of an actionable message or event that needs to be communicated to the operators. Carry out and then document a test of that procedure. Do 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 needs to be clear. The vendor often requires precise, detailed information such as serial numbers, machine type, and location.

- ▶ Batch procedures and scheduling

Most applications 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 need to be involved to create the necessary scheduling changes for the migrated application. The new schedules must 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 might change or be affected by the migration of the source application to the target application. Then, create a plan to perform the required operational migration tasks. And finally, run your plan.

5.8 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 need. 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:

- ▶ Disaster recovery (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.

- ▶ High availability (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.

5.8.1 Availability analysis

Migrating an application to a virtualized Linux environment on IBM LinuxONE 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. However, this analysis is not always straightforward. For example, test and development workloads are generally not considered to be mission-critical. However, because they might 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 might impact an application's availability, and unrelated events that might 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, and civil disturbances. These events can have a major impact on the ability of people to work.

Although this chapter focuses on the IT-related issues, you also create a plan to deal with the other, non-IT related events.

5.8.2 Single points of failure

In determining the DR requirements of an application, you need to look at the probability of failure of a component and the cost to eliminate a single point of failure (SPOF).

Table 5-3 lists the components of an IBM LinuxONE virtualized environment running an application as a Linux guest and the relative costs of rectifying a single point of failure.

Table 5-3 Potential single points of failure that can impact availability

Single point of failure	Probability of failure	Cost to rectify
LinuxONE hardware	Very low	High
LinuxONE LPAR	Very low	Low
LinuxONE hypervisor	Low	Low
Linux	Low	Very low
Disk system microcode	Low	Medium
Virtual network	Very low	Low
Physical network	Medium	Medium
Application	High	Very Low

Apart from hardware and software failures, the following types of planned outages can impact an application's availability:

- ▶ Hardware upgrades that require a power-on reset
- ▶ Configuration changes that require a reboot of the partition
- ▶ KVM or z/VM maintenance
- ▶ Linux kernel maintenance that requires a reboot
- ▶ Application maintenance

5.8.3 LinuxONE features for high availability

IBM LinuxONE was designed around providing HA. Perhaps the most design effort has gone in to the transparent recovery of processor errors. During 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 LinuxONE, 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 logical partitions (LPARs)
- ▶ Adding logical processors to a partition
- ▶ Adding logical channel subsystems (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 LinuxONE 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:

- ▶ A Customer Initiated Upgrade (CIU) is used for a permanent capacity upgrade.
- ▶ Capacity BackUp (CBU) is a predefined capacity for DR. A system at a DR site does not need to have the same capacity as the primary site. During 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 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 two hour increments that is available to be turned on to satisfy peak demand in workloads.

5.8.4 Availability scenarios

The following scenarios present a number of different situations where a LinuxONE 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.

All scenarios assume that the IBM LinuxONE is configured with redundant LPARs, redundant 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 LinuxONE System.

The application design needs to include redundant software servers. The storage infrastructure should also include redundant Fibre Channel switches, mirrored disks, and data.

Design the communications network around redundancy with redundant network routers, switches, hubs, and wireless access points.

For mission-critical systems, provide an uninterrupted power supply and 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, see 5.6, "Security analysis" on page 95.

Single LinuxONE LPAR: Clustered WebSphere Application Server

Figure 5-14 shows a LinuxONE partition that shares system resources to all Linux virtual machines in that partition. The WebSphere Application Servers are in a two-node cluster. In the unlikely event that a processor fails, IBM LinuxONE automatically switches the workloads to a spare or any unassigned processor without any disruption to the active task.

If a Linux virtual machine that runs the WebSphere Application Server workload fails, the other node in the cluster takes over if you are running WebSphere Application Server Network Deployment. This failover is achieved because an application deployed to a cluster runs on all members concurrently. Additional availability is provided through the nondisruptive addition of new virtual machines to the cluster.

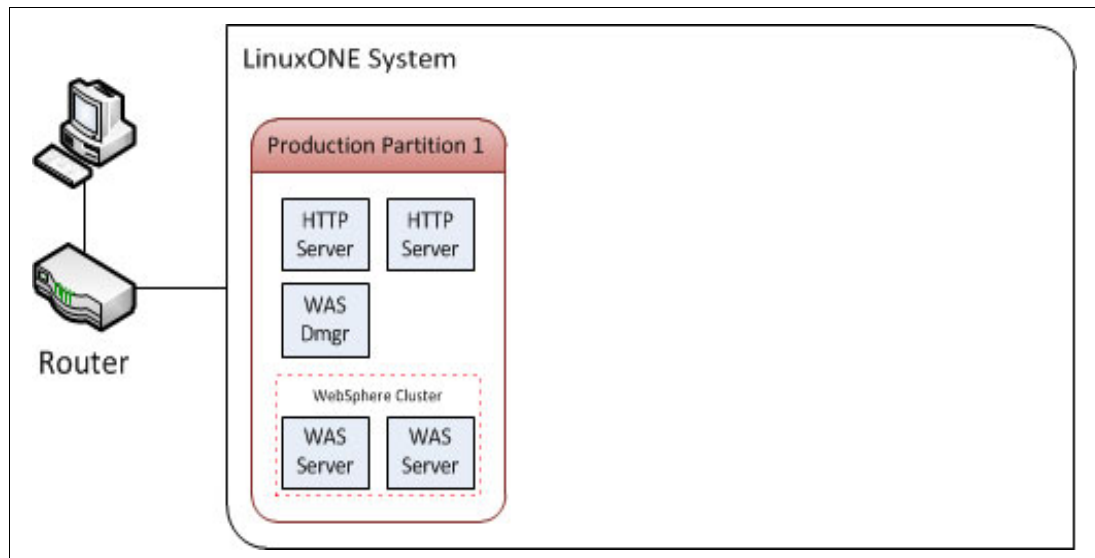


Figure 5-14 LinuxONE with a single partition running WebSphere Application Server cluster

This environment also provides extra availability through redundant HTTP servers.

Multiple LPARs: HA solution for Linux

Figure 5-15 shows a scenario with three partitions defined. Each partition can have one or more cores that are shared among partitions.

In this case, the production workload and WebSphere Application Server cluster is split across two LPARs, which give HA to WebSphere Application Server because a partition or hypervisor failure will not impact the availability of WebSphere Application Server.

Development and test workloads run in their own LPAR, so any errant servers have no impact on the production workloads. As in the first scenario, a failure of a LinuxONE processor is fixed automatically without any impact to the running application.

This configuration eliminates most failure points at a reasonable cost.

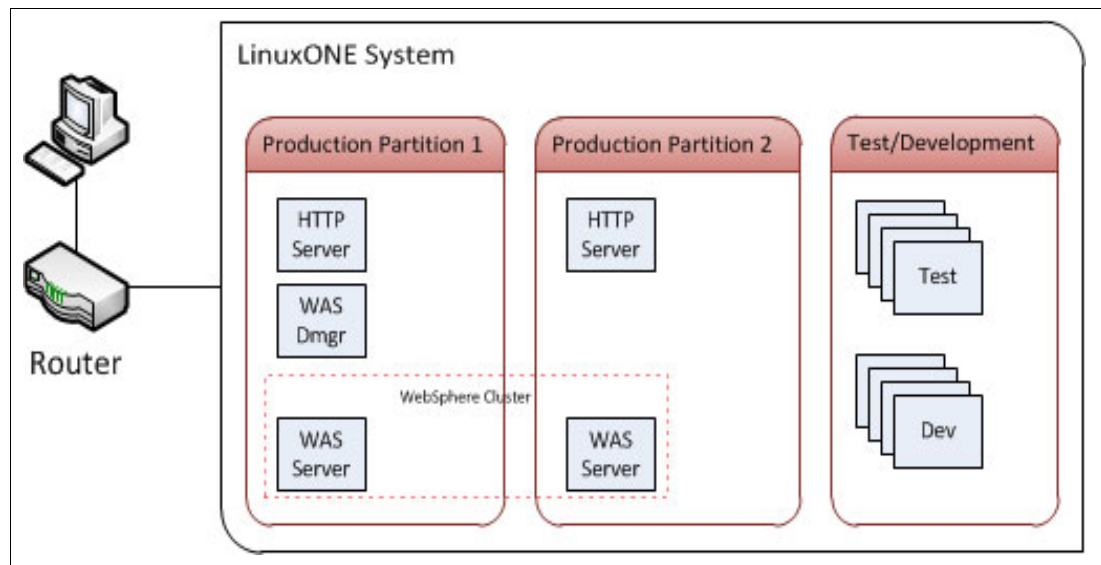


Figure 5-15 LinuxONE with multiple partitions running WebSphere Application Server cluster

Active/passive standby cluster with Pacemaker and Heartbeat

Beyond specialized facilities such as the WebSphere Application Server cluster described above, the Linux ecosystem has more generalized and robust clustering solutions available. A solution can be established with an open source product such as Pacemaker, or a commercial product such as Tivoli System Automation for Multiplatforms (SA MP). In this scenario, the virtual machines are established in an active/passive relationship. In this relationship, software runs on each server monitors and communicates with each other. The software coordinates an automated failover of traffic to the standby server during a failure of the primary system.

Figure 5-16 shows this configuration, where a single IP address (the “service IP”) acts as an alias to the network adapter of the primary partition.

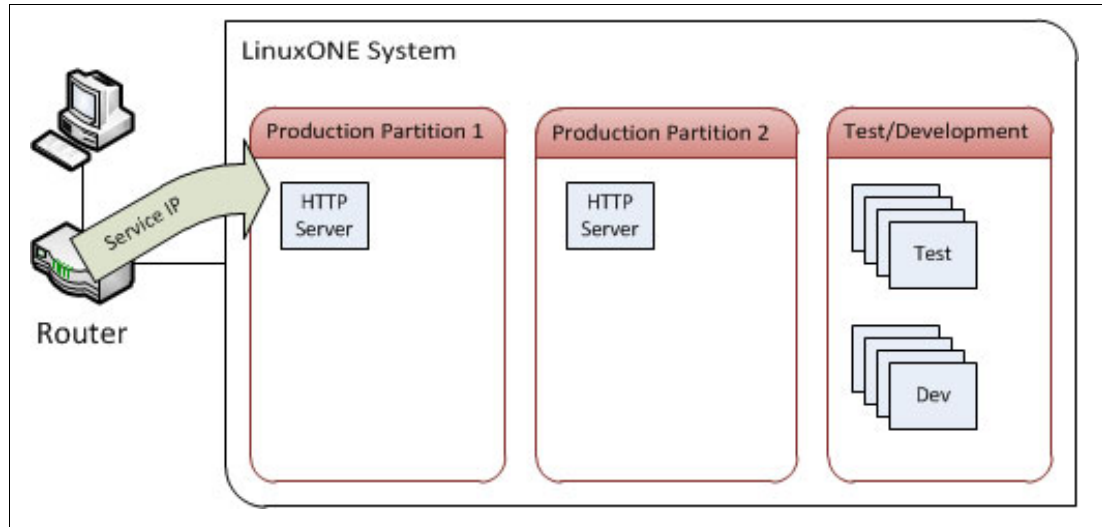


Figure 5-16 Active/passive cluster, during normal operation

At regular intervals, the clustering software verifies that the physical partitions, the virtual machines, and the server applications (a web server in this example) are all responsive. If any component is not, then the service IP points to the network of the passive system, as shown in Figure 5-17. The cluster can be configured as to what action is performed at this point, from notification to resource reboots of the failing system.

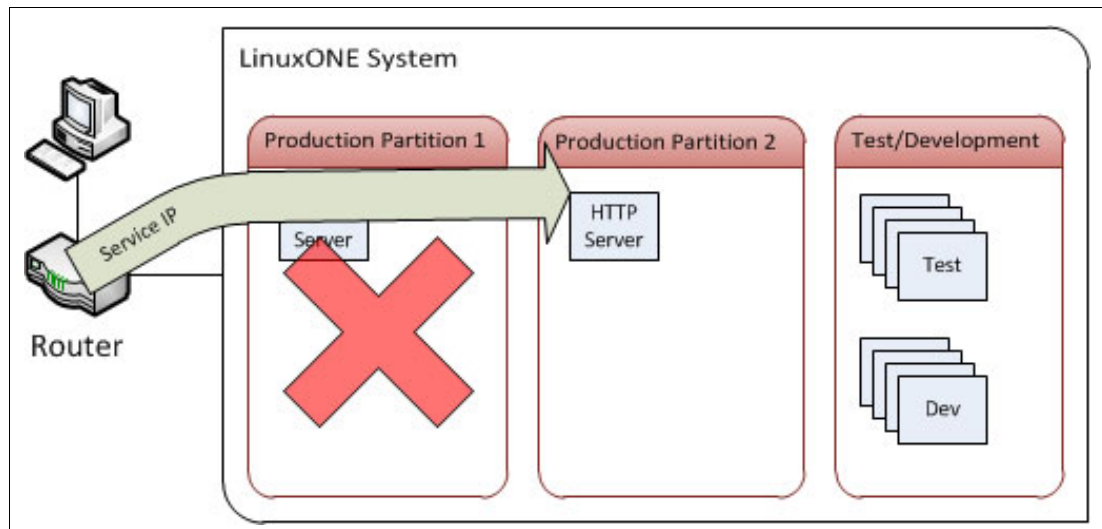


Figure 5-17 Active/passive cluster, after an outage occurs

For more information about Tivoli System Automation for Multiplatforms, see:

<http://www.ibm.com/software/products/en/category/system-workload-automation>

For more information about Pacemaker, see:

<http://www.clusterlabs.org>

Active/active application server cluster

Figure 5-18 shows a WebSphere Application Server setup in an active/active configuration where the WebSphere Application Server Cluster spans two Linux virtual machines in two partitions. This setup handles the very rare occurrence of the failure of a partition. More importantly, it also allows hypervisor maintenance to be performed without an outage to the WebSphere applications. This task would be scheduled for a time when the processing load is light. Guest relocation or migration can also be used to avoid an outage due to hypervisor maintenance.

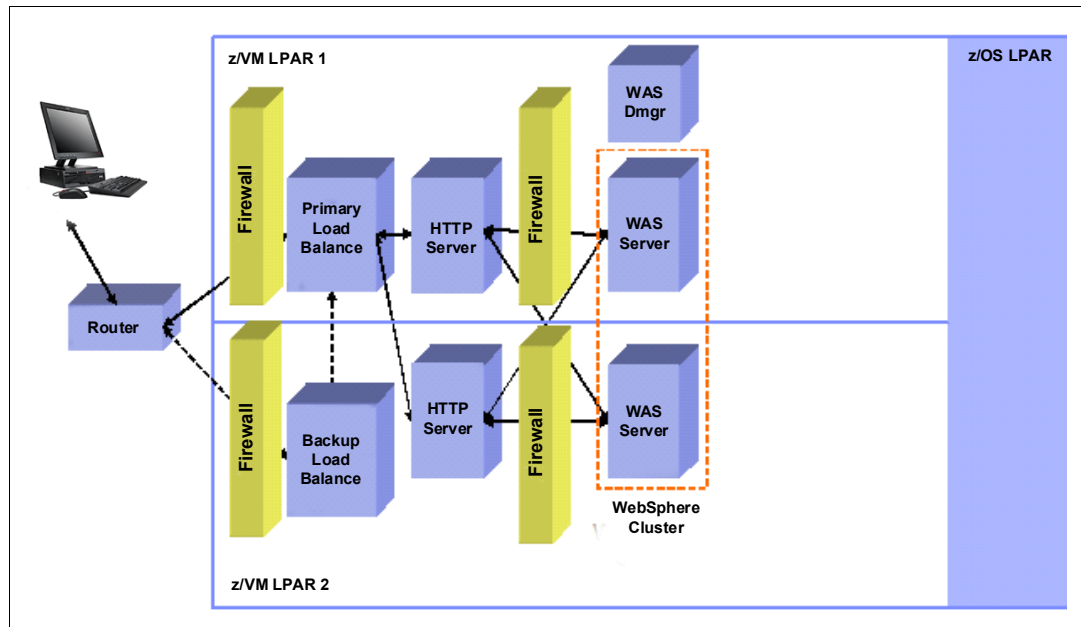


Figure 5-18 Active/active WebSphere Application Server cluster

Active/active application server cluster with database replication

Figure 5-19 on page 114 shows a DB2 database added to an active/active WebSphere cluster. To provide HA 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 (called *primary*) database to a target (called *standby*) database.

During a partition-wide outage of the primary DB2 system, the standby DB2 system takes over in seconds, thus providing high availability. Communication between the primary and standby systems is through TCP/IP, which in this case would be done by using the high-speed virtual network feature HiperSockets, available on LinuxONE.

The standby DB2 system can also be at a remote site to provide enhanced availability during a site failure.

IBM Tivoli SA MP running in both DB2 servers is designed to automatically detect a failure of the primary, and then issue commands on the standby for its DB2 to become the primary.

Other cluster management software can be used. However, SA MP and sample automation scripts are included with DB2 to manage the HA requirements of your DB2 database system.

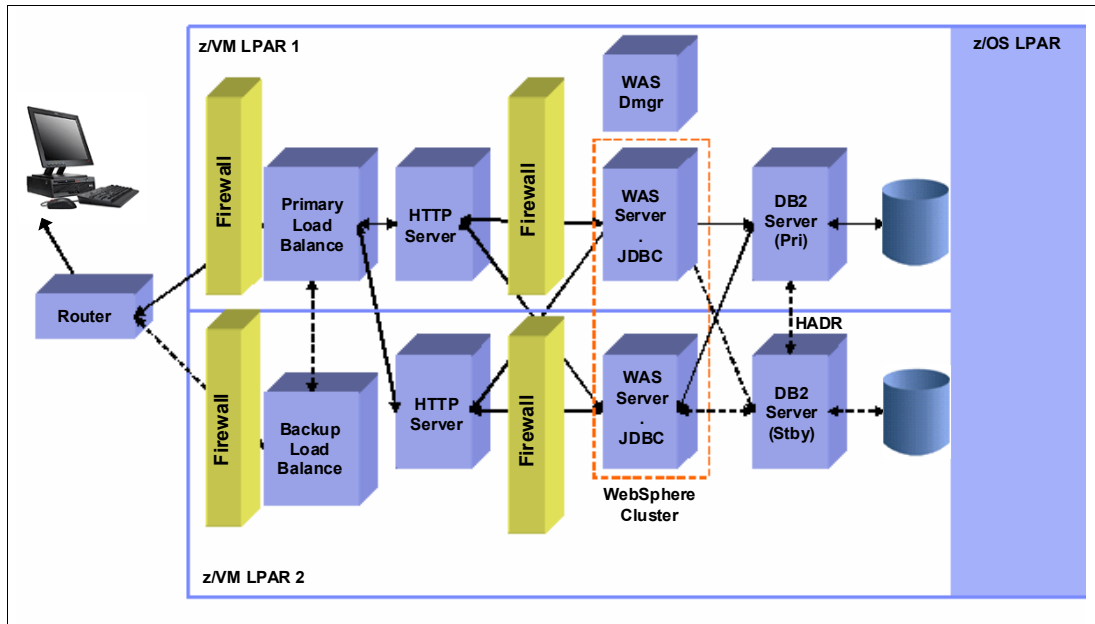


Figure 5-19 Active/active WebSphere Application Server cluster and DB2 HADR

Active/active application server cluster with database sharing

Figure 5-20 shows that database sharing was introduced by using Oracle Real Application Clusters (RACs). Oracle RAC provides HA for applications by having multiple RAC nodes share 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.

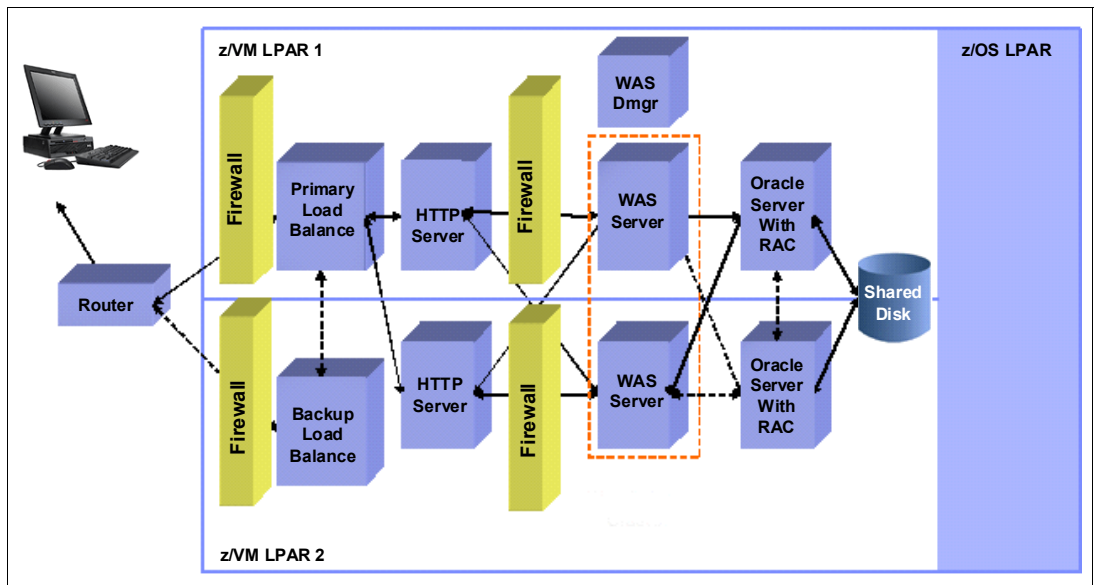


Figure 5-20 Database sharing using Oracle RAC

In a LinuxONE environment, communication between the database nodes uses a virtual LAN in the same LPAR or HiperSockets to other LPARs. Both methods are at memory-to-memory speeds with low latency.

For more information about Oracle RAC, see the following website:

<http://www.oracle.com>

Active/active cluster with database sharing across cities

For the ultimate availability solution, You can have two sites a metropolitan distance apart, while providing data sharing between clusters at each site. This configuration is achieved with the IBM Globally Dispersed Parallel Sysplex® (IBM GDPS®) Virtual Appliance. The GDPS Virtual Appliance supports both planned and unplanned outages, thus improving application availability and business continuity.

Distances greater than 100 km (62 miles) are also possible, although this configuration requires an asynchronous copy on the remote site, so it is not synchronized with the primary copy. For information about the GDPS Virtual Appliance, see *IBM GDPS Family: An Introduction to Concepts and Capabilities*, SG24-6374.

5.8.5 Linux-HA Project

The Linux-HA Project provides HA solutions for Linux through an open development community. Most Linux-HA software is licensed under the Free Software Foundation GNU General Public License (GPL) and the Free Software Foundation GNU Lesser General Public License (LGPL).

You can read more about Linux-HA project at their official website:

<http://www.linux-ha.org>

As the Linux-HA project has spun off individual components to address some limitations with its existing clustering support, with the primary cluster resource manager now being Pacemaker.

Note: For more information about Linux-HA and examples of its use, see *Achieving High Availability on Linux for System z with Linux-HA Release 2*, SG24-7711.

5.8.6 High Availability add-ons

Two solutions are worth your attention when you are looking for an HA third-party solution. Depending on the distribution that you have chosen for your environment, you might find add-ons that will facilitate the HA implementation for you, and help with installation, maintenance, and management.

For SUSE Linux Enterprise High Availability Extension, see the following site:

<http://www.suse.com/products/highavailability>

For Red Hat Enterprise Linux High Availability Add-On, see the following site:

<https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux>

5.8.7 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.

Service level agreements

To determine the availability requirements of applications that you want to migrate to LinuxONE, consider the needs of the business units that rely on these applications. Ideally, create SLAs that state requirements, such as availability needs, response time, maximum system utilization, and DR requirements. Use these SLAs as the basis for the design of the target system on Linux.

If SLAs do not exist, before designing a solution, discuss with the business units what levels of service you can offer and what level of investment that they are willing to make. The key to 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 SLAs are reviewed regularly because things will change.

According to IT Service Management principles, a service level agreement typically defines or covers 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: 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.
 - Definition of how availability is measured and who will do the measurement. For example, system availability, application availability, database availability, and network availability.
- ▶ Minimum system response time
 - Defined number and definition of where and how is it measured.

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 monitoring the average revenue that is generated in a specific amount of time. This amount provides an idea of the revenue that might be lost during an outage and how much you should spend to make the application more resilient. Other businesses 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.

5.9 Virtualized Environment to LinuxONE Cloud Migration

Today digital economy requires speed and agility to keeping up with competitive demands for new applications and big data analytics, as well maintaining existing infrastructure. Many of these enterprises also operate large vendor-dependent virtualized infrastructures to support their mission-critical, scale-up applications and enterprise databases. They value various products for their compute, network, storage, and management technologies. But they are also encountering new use cases that demand the agility that is offered by cloud.

This section covers some of the technical aspects that needs to be considered before migrating a distributed virtualized environment onto IBM LinuxONE Cloud. This process is usually easy if the workloads targeted for LinuxONE cloud migration have been virtualized. You cannot move the individual virtual machines directly on to a LinuxONE cloud, But you can create similar capabilities of the virtualized environment on the targeted LinuxONE cloud platform.

Before migration, create a chart of the following information for planning purposes:

- ▶ Performance and availability of a range of services and the linkages between them. Record performance metrics for each of the individual hypervisor.
- ▶ Application architectures and interdependencies.
- ▶ Network connectivity diagram for individual virtual machines and their applications.
- ▶ Security and isolation of networks among applications.
- ▶ Storage allocations, and their performance requirements.

5.9.1 Hypervisor Considerations

Different hypervisors have different degrees of guest and functionality support. For instance, in a OpenStack cloud platform, KVM and Xen Hypervisors are closely coupled with various network and storage support options. The following are the most important considerations:

- ▶ Standardization and hardening of virtual machines is very important. Before the migration, plan and validate an approach on how the virtual machines are going to be patched, maintained, and monitored.
- ▶ Validate that the security, management, and monitoring tools can be extended or similar functions are available in the targeted LinuxONE Cloud platform.
- ▶ Interoperability and isolation of applications functional interfaces and cloud services interfaces.
- ▶ Use LinuxONE based Workload patterns for deployment of standardized OS and middleware template for deployments.
- ▶ Analyze application portability on the targeted cloud platform and, if required, start application functional and performance testing activity on cloud.

5.9.2 Network considerations

Even though clouds platforms mask underlying differences at the infrastructure layer to allow for scale and dynamism, this homogeneity creates new network bandwidth and provisioning challenges.

Bandwidth requirements

In a traditional virtualized environment, physical servers are connected by physical switches, which meant that network administrators had total control of the network traffic and its security. When considering moving the workloads onto a LinuxONE based cloud, it is better to monitor the current bandwidth utilization. In a cloud environment, the virtual switch has links from the physical switch through the physical NIC that attaches to virtual machines. Therefore, it is better to validate and identify whether LinuxONE infrastructure network cards and internal virtual switch connectivity is sufficient, both in terms of throughput and latency.

Virtual switch and VLAN Tagging

In a LinuxONE Cloud Platform, virtualized networks also need a separation of VMs to ensure data privacy of one tenant in the cloud from another. Therefore, they need mechanisms to ensure that these networks can share a physical network link without compromising or leaking information between networks. To allow access to a physical network, most cloud automation software (like OpenStack) uses the virtual local area network (VLAN) tagging model. This approach requires network administrators to create pools of VLAN IDs on a physical switch. When a new VM or virtual application is created from the cloud software, a cloud consumer uses these VLAN IDs without having any delay.

5.9.3 Security considerations

The following are security consideration of the LinuxONE cloud service:

- ▶ Authentication and authorization of users and administrators of the cloud service.
- ▶ Configuration and operation of encryption both for data stored within the cloud service and also for data transmitted to and from the cloud service.
- ▶ Firewalls and configuration of other security capabilities.
- ▶ Virtual systems running on top of the hypervisor must have no means of network communication directly with the hypervisor.
- ▶ Hypervisor management traffic must use a separate physical network interface from the virtual guests.
- ▶ Create separate security zone spanning LPARs and Hypervisor depending on the application and organizational requirements.
- ▶ For KVM-based hypervisors, additional technologies like SELinux can be enforced in addition to security or firewall zones.
- ▶ Standardized user interfaces, APIs, protocols, and data formats are defined for cloud services.
- ▶ Open technologies are driving innovation, standardizing open source products for administration and business interfaces.
- ▶ Use existing or newer equivalent Access Management functions to authenticate and authorize access to virtual machines in cloud services.

5.9.4 LinuxONE cloud management

The following applications are available to help you with LinuxONE cloud management:

- ▶ OpenStack
- ▶ vRealize Automation for IBM LinuxONE

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 Manager Appliance (CMA) 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 IBM LinuxONE

With VMware vRealize Automation (vRA), you can extend your existing vRealize Automation cloud to IBM LinuxONE. vRealize Automation interfaces with the OpenStack enabled cloud endpoints from IBM LinuxONE to provide cloud management services. This configuration validates the openness of IBM LinuxONE systems to allow clients to use the same cloud management that is used within their distributed cloud environment (Figure 5-21).

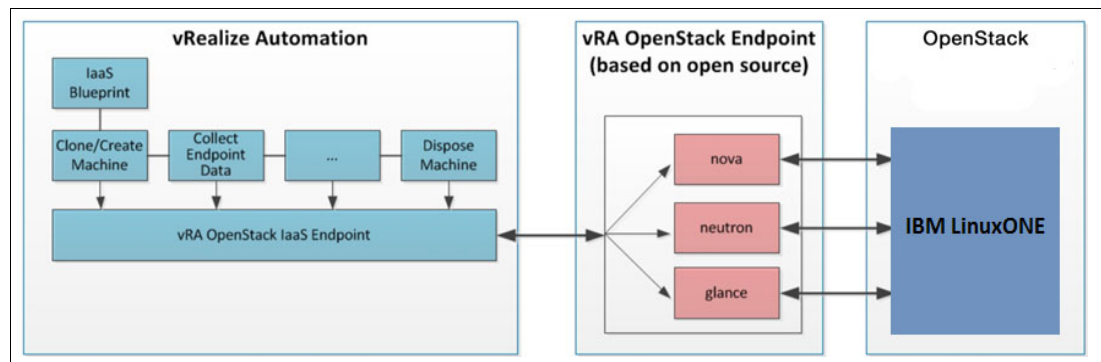


Figure 5-21 vRealize Automation with IBM LinuxONE framework



Hands-on migration

Now that migration planning and analysis have been completed, test your migration plan by performing a hands-on migration into 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, migrate from test to production.

Many methods of migrating your data from your source x86 server to your new LinuxONE server are available, and many ways of configuring your new LinuxONE environment.

This chapter describes a hands-on migration scenario performed in our lab where we migrated a WebSphere Application Server, DB2, and a simple load testing Java application called *Trade 6* from an x86 SUSE Linux Enterprise Server 11 SP3 environment to a LinuxONE environment.

This chapter includes the following sections:

- ▶ Setting up the system
- ▶ Migrating DB2 and its data
- ▶ Migrating the WebSphere Application Server

6.1 Setting up the system

This section describes the tasks required to create virtual guests on LinuxONE. We first completed our planning checklists to determine what resources were being used in the x86 environment. Next, we determined what to use in the LinuxONE environment, keeping performance in mind. We then used IBM Wave for z/VM to quickly and efficiently create the Linux guests in the partition. Finally, we performed the actual migration tasks.

6.1.1 Software products and tools checklist

Table 6-1 lists the software products.

Table 6-1 Software products and tools checklist for the x86 environment

Software products and tools checklist for x86 environment				
Name	Version	Vendor/Source website	License type	LinuxONE
DB2	10.5.0.4	IBM www.ibm.com		
WebSphere Application Server				

6.1.2 Hardware checklist

This section lists the hardware resources that are needed, either physical or virtual. In the checklist (Table 6-2) used in this project, the source environment's hardware resources were examined and those resources were cross referenced that with what could be used on LinuxONE.

Table 6-2 Hardware checklist for the x86 environment

Hardware planning checklist			
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	4	
Network connection^a			
Connection Description	Gigabit Ethernet	Gigabit Ethernet	
Connection Type	Gigabit Ethernet	Vswitch/GbE	
IP Address/Netmask	9.12.7.88/28	9.12.7.88/28	
Logical volumes:			
Volume Group OS : 20GB			
Volume Group DB : 150GB			
Volume Group WAS: 80GB			
Volume Group MGM: 20GB			

Hardware planning checklist			
Server name:			
Vlan number : Vswitch	2	2 : Vswitch1	
Disk Resource^b			
OS file system	/ : 30 : Ext3	/ : 2 :Ext4	Root
Mount Point : Size (in GB) : Type		/opt : 3 :Ext4 LV OS	Logical Volume
Mount Point : Size (in GB) : Type		/var : 5 :Ext4 LV OS	
Mount Point : Size (in GB) : Type		/var : 5 :Ext4 LV OS	
Mount Point : Size (in GB) : Type		/tmp : 1 :BRTFS LV OS	
DATA Filesystem			
Mount Point : Size (in GB) : Type	/DB : 100 : Ext3	/DB:100:Ext4 LV DB	Logical Volume
Mount Point : Size (in GB) : Type	/WAS : 50 : Ext3	/WAS:50:Ext4 LV WAS	
CUSTOM Filesystem			
Mount Point : Size (in GB) : Type		/MGM:10:Ext4 LV MGM	Logical Volume
Logical volumes: Volume Group OS : 20GB Volume Group DB : 150GB Volume Group WAS: 80GB Volume Group MGM: 20GB			

- a. For IBM LinuxONE, the available network connections are:
 - QETH
 - HiperSockets
 - Direct OSA-Express connection
- b. We used the Logical Volume Manager (LVM) for the Linux environment because it provides flexibility and reduces downtime of the environment with online resizing of the logical volumes

6.1.3 FCP and multipath

The failover configuration for FCP is not handled by PR/SM or z/VM, and must be done from within the Linux guest. Therefore, two N_Port ID Virtualization (NPIV) adapters must be attached to the guest system that is connected over the two different Fibre Channel fabrics. The description that follows is also applicable to a KVM environment because the multipathing must be performed by the host system, and then given as a single virtio device to the Linux guests of KVM.

The multipath setup itself is configured within the Linux 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 following command:

```
multipath -ll
```

Note regarding SUSE Linux Enterprise Server: The behavior of SCSI devices changed between SUSE Linux Enterprise Server 11 and SUSE Linux Enterprise Server 12. In SUSE Linux Enterprise Server 11, all LUNs had to be configured entirely manually. With SUSE Linux Enterprise Server 12, automatic LUN scanning has been added, and therefore all LUNs are automatically detected after the `zfc` host has been configured.

The actual configuration of multipath depends on the storage device used. For IBM DS8000® storage systems, the configuration shown in Example 6-1 can be used.

Example 6-1 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
  }
}
```

6.1.4 FCP migration setup tasks

The migration of Fibre Channel devices from a distributed system to LinuxONE with NPIV involves several different tasks:

1. Dedicate NPIV adapters

Assuming that the NPIV adapters have already been configured for LinuxONE, a pair of NPIV adapters attached to the two different zones must be dedicated to the new guest system. For redundancy, those adapters should come from two different physical cards:

- a. When using the KVM hypervisor, the NPIV adapters need to be configured on the KVM host as part of the normal SCSI configuration for a Linux system.
- b. When using the z/VM hypervisor, the NPIV adapters need to be assigned to the Linux guest for its configuration purposes.

Generally, dedicate the same virtual device (vdev) number to each guest. For example, if you configured two device ranges FA00-FA1D and FC00-FC1D for the two fabrics on partition LP1 and give real devices FA06 and FC06 to a guest, dedicate them as generic vdevs FA00 and FC00 as pairs:

```
DEDICATE FA00 FA06
DEDICATE FC00 FC06
```

With this configuration, the addresses that are seen by the guest 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 partition that will host your Linux guest (LP1 in the example that follows).
- b. Log on to the Support Element (SE) of the LinuxONE server, either through 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>** → **Configuration** → **FCP Configuration**. The **<system name>** is the name of the LinuxONE system.
- d. The easiest way to retrieve the information is to transfer the file with the WWNs for the partition to a remote FTP server by using 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 partition LP1, use the following command:
grep LP1 npiv-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, add the WWNs from step 2 to the respective fabric. If just one large zone exists, 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 WWN, whereas with NPIV the respective NPIV WWN is used. Normally, you can add several host connections to a specific storage group. This configuration 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 migration to the LinuxONE has completed, remove the old WWN from the host connection.

6.2 Migrating DB2 and its data

This section describe the migration of DB2 data from the source system (an x86 system named zs4p01-s1) to the target system (a LinuxONE partition named LNSUDB1). To perform the following steps, use the DB2 administrator user ID:

1. Copy the CREATE DATABASE command used to create the database on the source system to a file called CREATEDB_TRADE6DB.SQL and transfer it to the target system. This process can be done by using FTP or any FTP application such as Filezilla.
2. Edit the file locations to values that are applicable to the target system. Our environment used an iSCSI disk for the data called /db2_data. We want to use the same iSCSI disk on the new system. See 6.3, “Migrating the WebSphere Application Server” on page 127 for more information about switching the disk.

Example 6-2 shows our create database command. Do not run this command just yet.

Example 6-2 Command to create database

```
CREATE DATABASE TRADE6DB ON /db2_data
```

3. Use the DB2LOOK tool is used to extract the required Data Definition Language (DDL) statements that you need to reproduce the database objects of one database into another database. The tool can also generate the required SQL statements that are needed to replicate the statistics from the one database to the other, and the statements needed to replicate the database configuration, database manager configuration, and registry

variables. This information is important because the new database might not contain the exact same set of data as the original database, but you might still want the same access plans for the two systems. Use the DB2LOOK tool only on databases running on DB2 servers of Version 9.5 or later.

Generate the DDL by using the DB2LOOK command:

```
db2look -d trade6db -e -x -l -o trade6db.sql
```

where:

-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

-o: The name of the output file

For more information about the DB2LOOK tool, see the following site:

<https://ibm.biz/BdRJw6>

4. Send this file through FTP to your target location.
5. Because the **DB2 Backup** command cannot be used to move data between operating systems, use the DB2MOVE EXPORT command, as shown in Example 6-3. Ensure that the output of DB2MOVE is directed to an empty directory, as both an *.ixf file and an *.msg file are created for every table in the database.

Example 6-3 DB2MOVE export command

```
db2move trade6db export
```

6. While installing DB2 on LNSUDB1, we ran out of space. So we opened IBM Wave for z/VM, right-clicked the guest, and selected **More Actions** → **Manage Storage**. This process created a new partition in a matter of seconds.
7. On the target system, using the DB2 administrator user ID, run the file with the CREATE DATABASE command. Example 6-4 demonstrates how we ran our command by using the file created in Step 1.

Example 6-4 Running the create database command

```
db2 -tvf CREATEDB_TRADE6DB.SQL -z create db.log
```

8. Create the database objects by using the DDL file created by the DB2LOOK tool:

```
db2 -tvf trade6db.sql -z ddl.log
```

Note: This example assumes that you are migrating from a Linux on x86 to Linux in the LinuxONE environment. For specifics on migrating from MS Windows, see *Practical Migration to Linux on System z*, SG24-7727.

9. After the DDL file has been successfully processed, import the data into the database by using the DB2MOVE LOAD tool as shown:

```
db2 db2move trade6db load
```

Your database is now ready to be used.

6.3 Migrating the WebSphere Application Server

The process of migrating from one WebSphere Application Server environment to another is straightforward. The application was migrated by using the source system's WebSphere Administrative Console EXPORT command. This command creates an enterprise archive (EAR) file on the client running the WebSphere Application Server console using the Firefox web browser. We used the FileZilla FTP client to move the EAR file to the target system. The installation of the application on the target WebSphere Application Server running on Red Hat Enterprise Linux 6 (RHEL6) was undertaken by using the target system's WebSphere Administrative Console IMPORT command. Both the export and import actions completed without incident.



Post migration considerations

This chapter describes general post migration consideration concepts for getting acceptance, measuring performance, and tuning. Topics 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 will center around *performance* and *functionality*. IT organizations face these 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, you need to take some important steps before and after the migration implementation phase. This chapter includes the following sections:

- ▶ Gaining acceptance
- ▶ Performance measurement
- ▶ Performance tuning

7.1 Gaining acceptance

Migration projects are generally recognized as major changes to the IT environment. Each change requires significant testing and acceptance by various stakeholders. These stakeholders must decide whether the migration was a success.

Acceptance requires an understanding of the big picture before and after migration:

- ▶ Before the implementation phase starts, complete these tasks:
 - Decide and document test scope.
 - Decide and document test case (including test scenario).
 - Create post migration checklists for all components.
 - Collect performance data about the system.
 - Get acceptance from the stakeholders for testing.
- ▶ After the implementation is done, complete these tasks:
 - Use the post-migration checklists and check whether the implementation is complete.
 - Test the system by using documented test cases (complete and document all test scenarios).
 - Measure performance and compare it with the previous performance data.
 - If necessary, perform performance tuning.

Based on project scope and context, items used for acceptance testing can differ, but the following list is the most common acceptance tests that are performed before gaining stakeholder acceptance:

- ▶ Application testing
 - In some cases, usability testing might be required.
- ▶ Functional testing
- ▶ Performance testing
- ▶ Security testing
- ▶ User acceptance testing

7.2 Performance measurement

This section describes 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. Compare that data with the performance of the application on the target environment.

This section also covers monitoring commands and tools that can assist you in identifying and resolving performance inhibitors.

7.2.1 What is performance

“Performance” in computer systems is a relative term. Usually computer performance is described in measurable terms, such as transactions per second, response time, or time to process a specific task. 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 of the target system.

The initial performance of a new system is often not 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.

7.2.2 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 LinuxONE is that the application will not only be more resilient and available, 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 correct 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, the response time of a unit of work called a *transaction* is measured. This transaction can entail something as simple as checking an account balance, or 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 7-1 shows the various components that make up user response time.

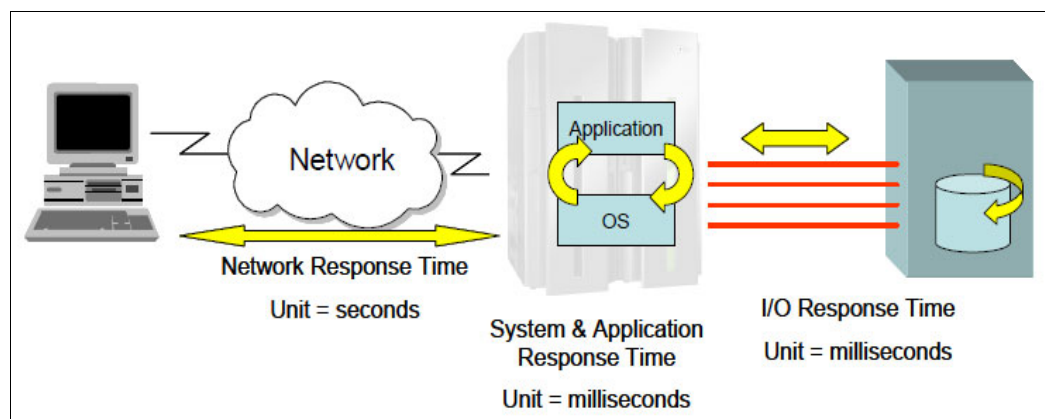


Figure 7-1 Components that make up the response time of transactions

Figure 7-1 shows that there are two points where response time can 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 shown in Figure 7-1 on page 131, the system response time includes 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 add 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, measure system response time on the source system, and assuming that the application has not changed greatly, measure the system response time on the target platform.

Transaction throughput

The transaction throughput performance metric might provide a more meaningful measure of system performance because it measures the number of transactions processed over a period of time. This period is typically one second, but can be any time period that you prefer.

In both cases, you have baseline performance metrics for the source system to properly compare the old and new systems.

7.3 Performance tuning

Tuning any system follows some basic principles because every hardware and software platform has unique features and characteristics that must be considered when you tune your environment. The art of performance tuning a system requires a strict combination of performance analyses, multi-step tuning processes, and change management.

Regardless of which tools you choose, the best methodology for analyzing the performance of a system is to start from the outside and work down to the small tuning details in the system. Gather data about overall health of the system hardware and processes. The following list is a sampling of the types of questions to 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 peaks 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 can 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. A system that is tuned for one type of workload often performs poorly with another type of workload. This consideration 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. 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 a hypervisor does. 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 by 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, formulates a hypothesis, tests that hypothesis by tuning or other means, and eventually solves the mystery. This process results in improved system performance. Bottleneck analysis and problem determination are facilitated by sophisticated tools such as IBM Tivoli OMEGAMON® XE on z/VM and Linux. These tools detect performance problems and alert a system administrator before degraded response time becomes evident.

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.



Part 3

Deployment

This part of the book describes deploying workloads and various applications to assist you during your deployment.

This part includes the following chapter:

- ▶ Chapter 8, “Deployment of workloads” on page 137



Deployment of workloads

This chapter covers how to deploy workloads to LinuxONE. 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 5.3, “Application analysis” on page 76, many workloads are a “good fit” on LinuxONE. Not all can be demonstrated in this book. The migration of some practical applications, such as IBM DB2, are illustrated as a hands-on exercise in Chapter 6, “Hands-on migration” on page 121. Mission critical applications, ERP, CRM, business intelligence, and more, are good to run on LinuxONE, but only generic examples can be included in a guide such as this. Your specific migration does not necessarily distill into a demonstration. Following the guides, the checklists, and the information in this book, and using this chapter of examples, will lead you to success.

Standard infrastructure applications are also well suited to the IBM LinuxONE, 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.

In this chapter, we provide examples of deploying workloads using High Availability clustering as well deploying the applications like MongoDB, Blockchain, MediaWiki and MySQL. We provide an example of deploying OpenLDAP, a central log server, and a file and print service.

This chapter includes the following sections:

- ▶ Deciding between containers and virtual machines
- ▶ Setting up Docker on Ubuntu 16.04 LTS
- ▶ Deploying MongoDB on IBM LinuxONE
- ▶ Deploying Hyperledger (Blockchain) Fabric on LinuxONE
- ▶ Deploying high availability clustering
- ▶ Deploying MediaWiki and MySQL
- ▶ Deploying OpenLDAP
- ▶ Deploying central log server
- ▶ Deploying Samba

8.1 Deciding between containers and virtual machines

The LinuxONE Platform is equipped with some of the fastest general-purpose processors in the world, ideally suited for data processing throughput. The large number of cores available in LinuxONE systems and their high input/output bandwidth mean that open source solutions can scale up and scale out.

Although the underlying hardware is ready for a highly scalable environment, there are advantages and disadvantages specific to having the solution on either a container or virtual machine. Containers can allow you to have a lot more applications in a single physical server than a virtual machine (VM) can. But we cannot fully negate the need for application deployments based on virtual machines.

The following are the deciding factors for determining whether the solution should be on either containers or virtual machines:

- ▶ How the application is going to be packaged. If you want to run multiple copies of a single app, say MongoDB, use containers. However, if you want the flexibility of running multiple applications (MongoDB with a Java based Homegrown Application), use a virtual machine.
- ▶ Usually containers tend to lock in to a particular version of an operating system and its subsystems. This feature can be an advantage for an administrator, because with containers you can create a portable, consistent operating environment for development, testing, and deployment. From a virtual machine 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.
- ▶ From a resource point of view, 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.
- ▶ 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 virtual machine.
- ▶ Concerning security, without any alterations to the container, a virtual machine is more secure than a container. Virtual machines have the advantage of having hardware isolation, whereas containers share kernel resources and application libraries. This feature means that if a virtual machine breaks down, it is less likely to affect other VMs in the same operating environment. For now, containers do not have hardware isolation. If your organization has high security requirements, stick with virtual machines.

Most organization will run both containers and VMs in their clouds and data-centers. The economy of containers at scale makes too much financial sense for anyone to ignore. At the same time, VMs still have their virtues. And LinuxONE provides the best in class features for running both containers and virtual machines.

8.2 Setting up Docker on Ubuntu 16.04 LTS

With the release of Docker version 1.11, the Docker Engine was heavily redesigned to make it compliant with the Open Container Initiative (OCI) run time. Specifically, it is built with the runC and containerd daemons.

- ▶ runC is a lightweight universal container run time. It is a command-line tool for creating and running containers according to the OCI specification.
- ▶ containerd is a daemon to control runC, built for performance and density. containerd uses runC's advanced features such as seccomp and user namespace support as well as checkpoint and restore for cloning and live migration of containers.

Docker Engine still does image management, and then passes the management of the image to containerd. Containerd then uses runC to run the container.

Ubuntu v16.04, is packaged with the pre-built Docker debian packages, and so containerd and runC are included by default. There are slight changes to the way that you deploy docker engine on the host system related to the packages being installed. The first step is to update the local host apt package index by using the command shown in Example 8-1.

Example 8-1 Update the Ubuntu package repository

```
blockchain@itsoblkchain:~$ sudo apt-get update
Hit:1 http://us.ports.ubuntu.com/ubuntu-ports xenial InRelease
Hit:2 http://us.ports.ubuntu.com/ubuntu-ports xenial-updates InRelease
Hit:3 http://us.ports.ubuntu.com/ubuntu-ports xenial-backports InRelease
Hit:4 http://ports.ubuntu.com/ubuntu-ports xenial-security InRelease
Reading package lists... Done
blockchain@itsoblkchain:~$
```

Use one of these options to deploy Docker packages onto the host system:

- ▶ Deploy the packages from official Docker engine distribution repository
- ▶ Deploy Docker Engine from the pre-packaged Ubuntu distribution itself

Example 8-2 shows deploying Docker using the second option. Ubuntu 16.04 LTS has pre-built debian packages for LinuxONE, which are named `docker.io` and `docker.registry`. The advantage of this option is that the docker daemon is integrated and tested on the distributions. However, the Docker Engine might be an older version or otherwise not the latest Docker versions.

Example 8-2 Install Docker packages

```
blockchain@itsoblkchain:~$ sudo apt-get -y install docker.io docker-registry
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils cgroupfs-mount containerd dns-root-data dnsmasq-base git git-man liberror-perl
  libnetfilter-contrack3 patch runc ubuntu-fan xz-utils
Suggested packages:
  mountall aufs-tools btrfs-tools debootstrap lxc rinse zfs-fuse | zfsutils git-daemon-run |
  git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb
  git-svn diffutils-doc
The following NEW packages will be installed:
  bridge-utils cgroupfs-mount containerd dns-root-data dnsmasq-base docker-registry docker.io
  git git-man liberror-perl libnetfilter-contrack3 patch runc ubuntu
```

```
0 upgraded, 15 newly installed, 0 to remove and 0 not upgraded.
Need to get 20.4 MB of archives.
After this operation, 134 MB of additional disk space will be used.
::
...
Processing triggers for systemd (229-4ubuntu10) ...
Processing triggers for ureadahead (0.100.0-19) ...
Processing triggers for libc-bin (2.23-0ubuntu3) ...
Processing triggers for dbus (1.10.6-1ubuntu3) ...
blockchain@itsoblkchain:~$
```

The Docker daemon binds itself to a UNIX socket instead of a TCP port, so Docker daemon always runs as a root user. To avoid having to use sudo when using docker commands, add the user to the docker group as shown in Example 8-3.

Example 8-3 Add users to docker group.

```
blockchain@itsoblkchain:~$ sudo usermod -aG docker blockchain
```

Log off and log in again for the user privilege to take effect. You can then verify the docker commands as shown in Example 8-4.

Example 8-4 Verification of docker version

```
blockchain@itsoblkchain:~$ docker --version
Docker version 1.11.2, build b9f10c9
blockchain@itsoblkchain:~$
```

As part of the verification, access the pre-built docker images part of the dockerhub and run a test hello-world container as shown in Example 8-5.

Example 8-5 Running docker containers

```
blockchain@itsoblkchain:~$ docker pull s390x/hello-world
Using default tag: latest
latest: Pulling from s390x/hello-world
9e1fa39565ec: Pull complete
Digest: sha256:ec5a634fcbd7db162961a3e3c32af3a85351cdb30b5cc593f59f8b5051f83c58
Status: Downloaded newer image for s390x/hello-world:latest

blockchain@itsoblkchain:~$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
s390x/hello-world   latest      0490c90ea2cf     3 months ago    1.984 kB

blockchain@itsoblkchain:~$ docker run 0490c90ea2cf

Hello from Docker on s390x!
This message shows that your installation appears to be working correctly.
```

You have successfully deployed docker.

8.3 Deploying MongoDB on IBM LinuxONE

MongoDB is an open source database that is considered to be the most popular and fastest 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 LinuxONE, it makes an ideal platform for scaling out and scaling up MongoDB based NoSQL workloads. This section looks at the steps for deploying MongoDB (as a Docker container) onto LinuxONE.

8.3.1 Work environment

This example uses Ubuntu 16.04 LTS as the host operating system for the MongoDB deployment. Because we have decided to install MongoDB as a docker container, the first step is to set up Docker on the host systems. For in-depth documentation 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 16.04 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 8-6 shows verifying the Docker configuration.

Example 8-6 Verification of Docker

```
itsoadmin@lnxmongo:~$ docker version
Client:
 Version:      1.12.1
 API version:  1.24
 Go version:   go1.6.3
 Git commit:   23cf638
 Built:        Mon, 19 Sep 2016 20:34:59 +1200
 OS/Arch:     linux/s390x
Server:
 Version:      1.12.1
 API version:  1.24
 Go version:   go1.6.3
 Git commit:   23cf638
 Built:        Mon, 19 Sep 2016 20:34:59 +1200
 OS/Arch:     linux/s390x
itsoadmin@lnxmongo:~$

itsoadmin@lnxmongo:~$ docker images
REPOSITORY TAG          IMAGE ID          CREATED          SIZE
itsoadmin@lnxmongo:~$
```

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.

Use the `docker search` command to search for repositories specific to a platform in Docker Hub as shown in Example 8-7. The command returns the pre-built docker images for LinuxONE from the public registry.

Example 8-7 Pre-built Docker Images for LinuxONE

```

itsoadmin@lnxmongo:~$ docker search brunswickheads/s390x
NAME                                DESCRIPTION                                STARS
OFFICIAL  AUTOMATED
brunswickheads/clefos71-base-s390x  Base ClefOS 7.1 Image                                2
brunswickheads/clefos71-java-s390x  Base IBM JDK 1.8 image for ClefOS 7.1 for ...        1
brunswickheads/nginx-1.8-s390x      nginx 1.8 container for s390x                        0
brunswickheads/bacula-fd-s390x      Bacula file director                                0
brunswickheads/openchain-build-s390x Utility container for building openchain-p...        0
brunswickheads/clefos71-mean-s390x  MEAN.JS is a full-stack JavaScript open-so...        0
brunswickheads/redis-2.8.19-s390x   redis server for ClefOS 7.2                          0
brunswickheads/amhub-s390x          Application Manager Docker Hub for ClefOS            0
brunswickheads/golang1.7-s390x      golang 1.7 - using the official go repo wh...        0
brunswickheads/mariadb-5.5-s390x    A MariaDB 5.5 container for Linux on z Sys...        0
brunswickheads/postgresql-9.2.13-s390x PostgreSQL 9.2.13 Container for ClefOS 7.1...        0
brunswickheads/mono-4.2.1-s390x     Mono development and runtime environment f...        0
brunswickheads/kubernetes-s390x     Kubernetes is a tool for orchestrating and...        0
brunswickheads/clefos72-base-s390x  A refresh of the ClefOS base image - upgra...        0
brunswickheads/spark-1.5.2-s390x    Apache Spark for ClefOS 7.1                          0
brunswickheads/golang1.5-s390x      golang 1.5.3 for s390x                              0
brunswickheads/golang-1.6-s390x     golang 1.6 for ClefOS on z Systems                   0
brunswickheads/registry-0.9.1-s390x docker registry for ClefOS 7.1                       0
brunswickheads/solr-6.1.0-s390x     Solr is the popular, blazing-fast, open so...        0
brunswickheads/hello-nodejs-s390x   Simple "Hello World" type app using NodeJS...        0
brunswickheads/saltmaster-s390x     Software to automate the management and co...        0
brunswickheads/mongodb-2.6.6-s390x  A MongoDB 2.6.6 container for Linux on z S...        0
brunswickheads/compose-ui-s390x     Docker Compose UI - web interface for Dock...        0
brunswickheads/clefos71-nodejs-s390x Base nodejs image for ClefOS 7.1 - Linux o...        0
brunswickheads/puppet-4.2.1-s390x   Puppet Server - 4.2.1                                0
itsoadmin@lnxmongo:~$

```

8.3.2 MongoDB container deployment

Now that you have a pre-built image for MongoDB from the Docker Hub, issue commands to `docker` to download and register the image to the local host system as shown in Example 8-8. These images are read-only snapshots of defined layers and commands.

Example 8-8 Downloading LinuxONE MongoDB Image from dockerhub

```

itsoadmin@lnxmongo:~$ docker pull brunswickheads/mongodb-2.6.6-s390x
Using default tag: latest
latest: Pulling from brunswickheads/mongodb-2.6.6-s390x
13e1ca791732: Pull complete
a3ed95caeb02: Pull complete

```

```
b380a104b128: Pull complete
a8260237e49a: Pull complete
Digest: sha256:03d725f66379a02c7cba2a8a9181843c2344b8bc7c8c1e64907ebeee05a3734f
Status: Downloaded newer image for brunswickheads/mongodb-2.6.6-s390x:latest
itsoadmin@lnxmongo:~$
```

Verify that the image has been correctly registered with the local Docker registry and allocated a local image ID as shown in Example 8-9.

Example 8-9 Verification of MongoDB docker image pull

```
itsoadmin@lnxmongo:~$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED
SIZE
brunswickheads/mongodb-2.6.6-s390x  latest      e00b2f10e8dd     11 months ago
408.7 MB
itsoadmin@lnxmongo:~$
```

When the MongoDB container was built, the directory `/mongodb/data` was used as a mount point for external storage, and it exposes ports 27017 and 28017. This technique allows connections from outside the container to access the mongodb container. Example 8-10 shows the configuration.

Example 8-10 Docker inspect

```
itsoadmin@lnxmongo:~$ docker inspect itsomongo
[
  {
    "Id": "ce2673ee3939527391518f7d721120284f8d263d72db2f626ba735032134dd19",
    "Created": "2016-10-10T14:37:04.508864Z",
    "Path": "/bin/sh",
    "Args": [
      "-c",
      "mongod --dbpath /mongodb/data --syslog --httpinterface --rest --smallfiles --noprealloc" ],
    "...",
    "Image": "brunswickheads/mongodb-2.6.6-s390x",
    "Volumes": {
      "/mongodb/data": {}
    },
  },
]
```

As in Example 8-10, the pre-built MongoDB container stores the data on the `/mongodb/data` folder on the host system. The idea is to create a data directory on the host system (outside 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 as shown in Example 8-11 on the host system.

Example 8-11 Create data directory

```
itsoadmin@lnxmongo:~$ sudo mkdir /mongodb
itsoadmin@lnxmongo:~$ sudo mkdir /mongodb/data
```

8.3.3 Running MongoDB container

Starting the MongoDB container by using the **docker run** command. Example 8-12 shows that the docker should instantiate the image named `brunswickheads/mongodb-2.6.6-s390x` and assign the newly instantiated container with the name `itsomongo`. 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 8-12 Starting MongoDB container

```
itsoadmin@lnxmongo:~$ docker run -p 27017:28017 --name itsomongo -d
brunswickheads/mongodb-2.6.6-s390x
ce2673ee3939527391518f7d721120284f8d263d72db2f626ba735032134dd19
```

8.3.4 Verifying and accessing MongoDB container

Check whether the container named `itsomongo` has started by using the **docker ps** command as shown in Example 8-13. 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 8-13 Container startup verification

```
itsoadmin@lnxmongo:~$ docker ps
CONTAINER ID        IMAGE                                     COMMAND                  CREATED
STATUS            PORTS                                     NAMES
ce2673ee3939      brunswickheads/mongodb-2.6.6-s390x    "/bin/sh -c 'mongod -"  4 seconds ago
seconds          27017/tcp, 0.0.0.0:27017->28017/tcp    itsomongo
itsoadmin@lnxmongo:~$
```

For in-depth information about the container, inspect the container by using the **sudo docker inspect <container id>** command.

You can use multiple 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 Mongo container and start a Mongo shell for creating a sample database (Example 8-14).

Example 8-14 Access MongoDB Container using Docker

```
itsoadmin@lnxmongo:~$ docker exec -it itsomongo /bin/bash
bash-4.2# mongo
MongoDB shell version: 2.6.6
connecting to: test
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
    http://docs.mongodb.org/
Questions? Try the support group
    http://groups.google.com/group/mongodb-user
```

```
> use mongodb
switched to db mongodb
> show dbs
admin (empty)
local 0.031GB
> db.itso.insert({"Redbook":"LinuxONE"})
WriteResult({ "nInserted" : 1 })
> show dbs
admin (empty)
local 0.031GB
mongodb 0.031GB
>
```

This method provides a quick way to have a highly scalable environment to work on for any solution involving MongoDB containers that are deployed on LinuxONE.

8.4 Deploying Hyperledger (Blockchain) Fabric on LinuxONE

With the open source Linux operating system comes a wide variety of open source applications. The latest innovation in open source is Blockchain. Blockchain is a specific type of network over which members track and exchange digitized assets. A shared ledger contains the single record of all network transactions, and is replicated across all network members. Chaincode applications contain self-executing contracts and client-side applications that interface with the network through an SDK or API.

Because LinuxONE is the most secured platform, it allows users to innovate with DevOps by providing purpose-built, cutting-edge solutions around open source technologies. LinuxONE provides superior security with performance and sustainable higher transaction encoding speeds. Whether generating digital signatures or hashing encryption for the chain, LinuxONE uses cryptographic accelerators.

This section looks at deploying Hyperledger Fabric on a LinuxONE environment.

8.4.1 Environment

This section describes how to set up a self-contained environment for application development with the Hyperledger Fabric. This example uses Ubuntu 16.04 LTS on LinuxONE as the host operating system for deploying Hyperledger fabric on LinuxONE.

Two methods can be used to deploy Hyperledger and its security services:

- ▶ Deploy Hyperledger Fabric and member services as daemons on the host systems.
- ▶ Deploy Hyperledger Fabric and its member services as Docker containers on the host system.

This section shows deploying the Hyperledger Fabric as a Docker container. Hyperledger fabric requires the following components to be installed:

- ▶ The Golang programming language
- ▶ RocksDB
- ▶ Hyperledger Fabric:
 - peer daemon
 - Membership and security services

Docker deployment

Hyperledger fabric relies on Docker for deploying and starting chaincodes on the peer nodes. Docker deployment is covered in 8.2, “Setting up Docker on Ubuntu 16.04 LTS” on page 139 and this section assumes that Docker has successfully been deployed and configured. As stated before, Docker requires the aufs package to use the storage driver for its management. Example 8-15 shows the docker deployment verification command.

Example 8-15 Docker deployment verification

```
itsoadmin@itsoblkchain:~$ sudo usermod -a -G docker itsoadmin

itsoadmin@itsoblkchain:~$ docker --version
Docker version 1.12.1, build 23cf638
itsoadmin@itsoblkchain:~$
```

Example 8-16 shows the command that is used to verify the ports that are used by the Docker container.

Example 8-16 Updating the Docker parameter file with the ports.

```
itsoadmin@itsoblkchain:~$ netstat -ant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 129.40.119.76:22       9.187.63.135:59583     ESTABLISHED
tcp        0      0 129.40.119.76:22       9.187.63.135:59819     ESTABLISHED
tcp        0      0 172.18.0.1:52162       172.18.0.2:7051       ESTABLISHED
tcp6       0      0 :::7050                 :::*                     LISTEN
tcp6       0      0 :::7051                 :::*                     LISTEN
tcp6       0      0 :::7053                 :::*                     LISTEN
tcp6       0      0 :::7054                 :::*                     LISTEN
tcp6       0      0 :::22                   :::*                     LISTEN
tcp6       0      0 :::5000                 :::*                     LISTEN
```

Docker Compose

For deploying Hyperledger containers, use the **docker -compose** command to run the hyperledger and membersvc images and spin up a local environment, as shown in Example 8-17.

Example 8-17 Docker compose version

```
root@Blockchain:~# docker-compose version
docker-compose version 1.8.1, build 878cff1
docker-py version: 1.10.6
CPython version: 2.7.12
OpenSSL version: OpenSSL 1.0.2g  1 Mar 2016
root@Blockchain:~#
```

Clone the image

Use the git command as shown in Example 8-18 to clone the fabric images from GitHub. If you do not have Git installed, download the appropriate Git client for your operating system.

Example 8-18 Clone fabric images

```
root@Blockchain:~# git clone https://github.com/IBM-Blockchain/fabric-images.git
Cloning into 'fabric-images'...
remote: Counting objects: 181, done.
remote: Total 181 (delta 0), reused 0 (delta 0), pack-reused 181
Receiving objects: 100% (181/181), 30.75 KiB | 0 bytes/s, done.
Resolving deltas: 100% (84/84), done.
Checking connectivity... done.
root@Blockchain:~# ls
```

From the docker-compose folder, verify and validate the presence of the files listed in Example 8-19. The docker-compose.yaml file located in the docker-compose directory has the commands to retrieve the relevant Docker container images.

Example 8-19 Review the git clones files

```
root@Blockchain:/opt# cd fabric-images/
root@Blockchain:/opt/fabric-images# ls -la
total 48
drwxr-xr-x 4 root root 4096 Nov  8 19:23 .
drwxr-xr-x 3 root root 4096 Nov  8 19:24 ..
drwxr-xr-x 8 root root 4096 Nov  8 19:23 .git
-rw-r--r-- 1 root root 11357 Nov  8 19:23 LICENSE
-rw-r--r-- 1 root root 12698 Nov  8 19:23 README.md
drwxr-xr-x 4 root root 4096 Nov  8 19:23 docker-compose
-rw-r--r-- 1 root root 3458 Nov  8 19:23 v0.6_migration.md
root@Blockchain:/opt/fabric-images#
```

Setting the environment

In the docker-compose folder, the setenv.sh file is used to set the environment variables pertaining to the specific platform where the docker images are going to be downloaded and spun off. Set the environment by running the setenv.sh script, Example 8-20 lists the contents of the setenv.sh file and then runs the file.

Example 8-20 Setting the environment

```
root@Blockchain:/opt/fabric-images/docker-compose# cat setenv.sh
#!/bin/bash

arch=`uname -m`

case $arch in
"x86_64")
    export ARCH_TAG="x86_64-0.6.1-preview"
    ;;
"s390x")
    export ARCH_TAG="s390x-0.6.1-preview"
    ;;
"ppc64le")
    export ARCH_TAG="ppc64le-0.6.1-preview"
    ;;
;
```

```

*)
  echo "No Architectural Images Available for Architecture: $arch - Please call ibm service"
  return
;;
esac

```

```

cat baseimage/Dockerfile.in | sed -e "s/_ARCH_TAG_/$ARCH_TAG/g" > baseimage/Dockerfile
root@Blockchain:/opt/fabric-images/docker-compose#

```

```

root@Blockchain:/opt/fabric-images/docker-compose# . setenv.sh

```

Deploying Hyperledger docker images

After the environment variables are set up for the deployment of hyperledger containers, issue the **docker-compose** command as shown in Example 8-21.

Example 8-21 Hyperledger codebase clone from GitHub

```

root@Blockchain:/opt/fabric-images/docker-compose# docker-compose -f single-peer-ca.yaml up
Creating network "dockercompose_default" with the default driver
Pulling membersrv (ibmblockchain/fabric-membersrv:s390x-0.6.1-preview)...
s390x-0.6.1-preview: Pulling from ibmblockchain/fabric-membersrv
93998b6341dd: Pull complete
49efd449046a: Pull complete
190419188b72: Pull complete
e1ac91a020e2: Pull complete
de863481df23: Pull complete
8e57d3e8877a: Pull complete
5ccbd057c15e: Pull complete
f643ea2d86f3: Pull complete
a12b60e4f7f1: Pull complete
5c8487fe8680: Pull complete
e8d30cc6981c: Pull complete
Digest: sha256:b6d3df0b3e68649b6e686bf119a15575b3ba97977f981a2a3f31af2ce2f5adef
Status: Downloaded newer image for ibmblockchain/fabric-membersrv:s390x-0.6.1-preview
Pulling vp (ibmblockchain/fabric-peer:s390x-0.6.1-preview)...
s390x-0.6.1-preview: Pulling from ibmblockchain/fabric-peer
93998b6341dd: Already exists
49efd449046a: Already exists
190419188b72: Already exists
e1ac91a020e2: Already exists
de863481df23: Already exists
8e57d3e8877a: Already exists
5ccbd057c15e: Already exists
f643ea2d86f3: Already exists
a12b60e4f7f1: Already exists
5c8487fe8680: Already exists
0cffd252d196: Pull complete
Digest: sha256:823200e31bf6f5baad9f92c4c43e82a089e9fcdc820b9827e032103466f2a63b
Status: Downloaded newer image for ibmblockchain/fabric-peer:s390x-0.6.1-preview
Building baseimage
Step 1 : FROM ibmblockchain/fabric-peer:s390x-0.6.1-preview
----> 7412ad951528
Successfully built 7412ad951528
WARNING: Image for service baseimage was built because it did not already exist. To rebuild this
image you must use `docker-compose build` or `docker-compose up --build`.

```

```
Creating dockercompose_baseimage_1
Creating dockercompose_membersrv_1
Creating dockercompose_vp_1
Attaching to dockercompose_baseimage_1, dockercompose_membersrv_1, dockercompose_vp_1
```

The **docker-compose** command automatically starts validating peer nodes and membersrv services. In addition to the peer and membersrv executable files, supporting Docker images are created for development use (see Example 8-22).

Example 8-22 Hyperledger fabric and membersrv execution logs

```
membersrv_1 | 11:39:24.946 [server] main -> INFO 001 CA Server
(0.6.1-preview-snapshot-e4a9b47)
membersrv_1 | 11:39:24.954 [ca] NewCA -> INFO 002 Fresh start; creating databases, key pairs,
and certificates.
membersrv_1 | 11:39:25.111 [eca] Start -> INFO 003 Starting ECA...
membersrv_1 | 11:39:25.111 [eca] startECAP -> INFO 004 ECA PUBLIC gRPC API server started
membersrv_1 | 11:39:25.111 [eca] startECAA -> INFO 005 ECA ADMIN gRPC API server started
membersrv_1 | 11:39:25.111 [eca] Start -> INFO 006 ECA started.
membersrv_1 | 11:39:25.111 [tca] Start -> INFO 007 Starting TCA services...
membersrv_1 | 11:39:25.111 [tca] startTCAP -> INFO 008 TCA PUBLIC gRPC API server started
membersrv_1 | 11:39:25.111 [tca] startTCAA -> INFO 009 TCA ADMIN gRPC API server started
membersrv_1 | 11:39:25.111 [tca] Start -> INFO 00a TCA started.
membersrv_1 | 11:39:25.111 [tlsca] Start -> INFO 00b TLSCA started.
dockercompose_baseimage_1 exited with code 0
vp_1 | 11:39:35.271 [logging] LoggingInit -> DEBU 001 Setting default logging level to
DEBUG for command 'node'
vp_1 | 11:39:35.271 [peer] func1 -> INFO 002 Auto detected peer address:
172.18.0.2:7051
vp_1 | 11:39:35.271 [peer] func1 -> INFO 003 Auto detected peer address:
172.18.0.2:7051
vp_1 | 11:39:35.272 [eventhub_producer] AddEventType -> DEBU 004 registering BLOCK
vp_1 | 11:39:35.272 [eventhub_producer] AddEventType -> DEBU 005 registering CHAINCODE
vp_1 | 11:39:35.273 [eventhub_producer] AddEventType -> DEBU 006 registering REJECTION
vp_1 | 11:39:35.273 [eventhub_producer] AddEventType -> DEBU 007 registering REGISTER
vp_1 | 11:39:35.273 [nodeCmd] serve -> INFO 008 Security enabled status: true
vp_1 | 11:39:35.273 [nodeCmd] serve -> INFO 009 Privacy enabled status: false
vp_1 | 11:39:35.273 [db] open -> DEBU 00a Is db path [/var/hyperledger/production/db]
empty [true]
vp_1 | 11:39:35.273 [db] open -> INFO 00b Setting rocksdb maxLogFileSize to 10485760
vp_1 | 11:39:35.274 [db] open -> INFO 00c Setting rocksdb keepLogFileNum to 10
vp_1 | 11:39:35.286 [nodeCmd] func1 -> DEBU 00d Registering validator with enroll ID:
vp
vp_1 | 11:39:35.286 [crypto] RegisterValidator -> INFO 00e Registering validator [vp]
with name [vp]...
vp_1 | 11:39:35.287 [eventhub_producer] start -> INFO 00f event processor started
```

Docker containers are used by the Hyperledger Fabric peer component when deploying Chaincode. The peer communicates with the docker daemon to initially create docker images (Example 8-23) based on the Golang toolchain docker image and contains the compiled chaincode built from the source specified by the peer chaincode deploy command. Docker containers are started by the peer and are executed whenever the transactions are , e.g., invoked or queried.

Example 8-23 Docker images downloaded by docker compose

```
itsoadmin@Blockchain:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ibmblockchain/fabric-membersvc	s390x-0.6.1-preview	ac31f05c9853	3 weeks ago	1.679 GB
hyperledger/fabric-baseimage	latest	7412ad951528	3 weeks ago	1.686 GB
ibmblockchain/fabric-peer	s390x-0.6.1-preview	7412ad951528	3 weeks ago	1.686 GB
s390x/hello-world	latest	0490c90ea2cf	4 months ago	1.984 kB

```
itsoadmin@Blockchain:~$
```

Hyperledger fabric and membersvc verification

Now that the hyperledger and membership services are deployed, verify the environment by logging in to the system and then deploying a simple example chaincode. Example 8-24 shows a test login to the hyperledger peer container and issuing the login command with user credentials.

Example 8-24 Log in test_user to deploy a chaincode

```
itsoadmin@Blockchain:~$ docker exec -it dockercompose_vp_1 bash
root@cc37eb03a9ba:/opt/gopath/src/github.com/hyperledger/fabric# peer network login test_user0
-p MS9qrN8hFj1E
11:55:26.356 [logging] LoggingInit -> DEBU 001 Setting default logging level to DEBUG for
command 'network'
11:55:26.356 [networkCmd] networkLogin -> INFO 002 CLI client login...
11:55:26.356 [networkCmd] networkLogin -> INFO 003 Local data store for client loginToken:
/var/hyperledger/production/client/
11:55:26.356 [networkCmd] networkLogin -> INFO 004 Logging in user 'test_user0' on CLI
interface...
11:55:26.391 [networkCmd] networkLogin -> INFO 005 Storing login token for user 'test_user0'.
11:55:26.391 [networkCmd] networkLogin -> INFO 006 Login successful for user 'test_user0'.
11:55:26.391 [main] main -> INFO 007 Exiting.....
root@cc37eb03a9ba:/opt/gopath/src/github.com/hyperledger/fabric#
```

After the user ID is successfully logged in, deploy an example chaincode from the samples packaged with the hyperledger. Example 8-25 shows deploying a simple chaincode by passing it first, the userid (because security is enabled) by using the `-u` parameter, then the path by using the `-p` parameter and finally, a constructor by using the `-c` parameter. The constructor describes what function to trigger to initialize the chaincode state upon deployment. In this case, the deploy transaction initializes the chaincode by running a target initializing function and passing in some parameters.

Example 8-25 Deploying chaincode on the peer node.

```
root@cc37eb03a9ba:/opt/gopath/src/github.com/hyperledger/fabric# peer chaincode deploy -u
test_user0 -p github.com/hyperledger/fabric/examples/chaincode/go/chaincode_example02 -c
'{"Args": ["init", "a", "100", "b", "200"]}'
12:01:25.748 [logging] LoggingInit -> DEBU 001 Setting default logging level to DEBUG for
command 'chaincode'
12:01:25.748 [chaincodeCmd] getChaincodeSpecification -> INFO 002 Local user 'test_user0' is
already logged in. Retrieving login token.
```

```
12:01:27.265 [chaincodeCmd] chaincodeDeploy -> INFO 003 Deploy result: type:GOLANG
chaincodeID:<path:"github.com/hyperledger/fabric/examples/chaincode/go/chaincode_example02"
name:"ee5b24a1f17c356dd5f6e37307922e39ddba12e5d2e203ed93401d7d05eb0dd194fb9070549c5dc31eb63f4e65
4dbd5a1d86cbb30c48e3ab1812590cd0f78539" > ctorMsg:<args:"init" args:"a" args:"100" args:"b"
args:"200" >
Deploy chaincode:
ee5b24a1f17c356dd5f6e37307922e39ddba12e5d2e203ed93401d7d05eb0dd194fb9070549c5dc31eb63f4e654dbd5a
1d86cbb30c48e3ab1812590cd0f78539
12:01:27.265 [main] main -> INFO 004 Exiting.....
root@cc37eb03a9ba:/opt/gopath/src/github.com/hyperledger/fabric#
```

After the successful deployment of the chaincode, start a transaction on the Blockchain environment as shown in Example 8-26.

Example 8-26 Start chaincode

```
root@cc37eb03a9ba:/opt/gopath/src/github.com/hyperledger/fabric#peer chaincode invoke -u
test_user0 -n
ee5b24a1f17c356dd5f6e37307922e39ddba12e5d2e203ed93401d7d05eb0dd194fb9070549c5dc31eb63f4e654dbd5a
1d86cbb30c48e3ab1812590cd0f78539 -c '{"Args": ["invoke", "a", "b", "10"]}'
```

```
12:04:39.128 [logging] LoggingInit -> DEBU 001 Setting default logging level to DEBUG for
command 'chaincode'
12:04:39.128 [chaincodeCmd] getChaincodeSpecification -> INFO 002 Local user 'test_user0' is
already logged in. Retrieving login token.
12:04:39.273 [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 003 Successfully invoked transaction:
chaincodeSpec:<type:GOLANG
chaincodeID:<name:"ee5b24a1f17c356dd5f6e37307922e39ddba12e5d2e203ed93401d7d05eb0dd194fb9070549c5
dc31eb63f4e654dbd5a1d86cbb30c48e3ab1812590cd0f78539" > ctorMsg:<args:"invoke" args:"a" args:"b"
args:"10" > secureContext:"test_user0" > (758462cb-fdd5-48b7-b4b5-04d97d5733d1)
12:04:39.273 [main] main -> INFO 004 Exiting.....
```

Example 8-26 shows starting a simple transaction using the example chaincode. In Example 8-27, query the environment to see whether the previously run transaction has completed successfully.

Example 8-27 Query transaction.

```
root@cc37eb03a9ba:/opt/gopath/src/github.com/hyperledger/fabric#peer chaincode query -u
test_user0 -n
ee5b24a1f17c356dd5f6e37307922e39ddba12e5d2e203ed93401d7d05eb0dd194fb9070549c5dc31eb63f4e654dbd5a
1d86cbb30c48e3ab1812590cd0f78539 -c '{"Args": ["query", "a"]}'
12:04:52.321 [logging] LoggingInit -> DEBU 001 Setting default logging level to DEBUG for
command 'chaincode'
12:04:52.321 [chaincodeCmd] getChaincodeSpecification -> INFO 002 Local user 'test_user0' is
already logged in. Retrieving login token.
12:04:52.470 [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 003 Successfully queried transaction:
chaincodeSpec:<type:GOLANG
chaincodeID:<name:"ee5b24a1f17c356dd5f6e37307922e39ddba12e5d2e203ed93401d7d05eb0dd194fb9070549c5
dc31eb63f4e654dbd5a1d86cbb30c48e3ab1812590cd0f78539" > ctorMsg:<args:"query" args:"a" >
secureContext:"test_user0" >
Query Result: 90
12:04:52.470 [main] main -> INFO 004 Exiting.....
```

8.5 Deploying high availability clustering

Both Red Hat Enterprise Linux Servers and SUSE Linux Enterprise Servers deliver add-on products that provide high availability (HA) failover clustering for their respective Linux operating systems. These products can be deployed to increase the availability and reliability of workloads on LinuxONE. During a catastrophic event that takes down one cluster member, the applications running on that member can be automatically brought online on one of the other members of the cluster, with no perceived downtime. SUSE Linux Enterprise Server High Availability Extensions even offers geodistribution clustering, enabling the restart of a database or application from a remote site.

8.6 Deploying MediaWiki and MySQL

A 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 LinuxONE.

The Linux environment on x86 is largely the same as it is on LinuxONE, with a few notable exceptions. Configuration files on the x86 are in the same place on your Linux images on LinuxONE, unless you deliberately choose to keep them in a different place. Hence, the MySQL configuration files, for example, typically only need to be copied from the x86 server to the LinuxONE server and placed in the same location in the file system (`/etc/my.cnf`).

Migrating to LinuxONE should be tested first in a test environment before performing the migration to the production environment.

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 from the LUN where the MySQL database is stored.

In today's data centers, it is common to store application data on external disks in this way. This practice makes it much easier to move (migrate) services from one host to another. It commoditizes the operating system and the various services. This is a preferred practice in the industry for these tasks:

- ▶ Maintaining the configurations of the operating system and services using a configuration management tool
- ▶ Deploying the operating system and services configurations out to virtual machines either by using a cloud tool or an external application such as Puppet, Chef, or caffeine-hx

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, the application data is stored on external disks. Although it is possible to back up the data, transfer the data from one host to another, and reimport the data into the new running service, this method is very slow. The method demonstrated here is a much faster way to accomplish migration tasks.

8.6.1 Analysis and planning

Following the guidelines and recommendations outlined in Chapter 4, “Migration process” on page 43, and Chapter 5, “Migration analysis” on page 55, 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 scenario, the Linux image has already been set up and a minimal Linux operating system installed. The Linux guest is called LNSUDB2 and is running SUSE Linux Enterprise Server 11 SP3, with one virtual CPU and 1 GB of virtual memory. It is assumed that an adequate package management (RPM) repository for installation source is already set up and available for the installation of the application software.

8.6.2 Installing the LAMP stack

The installation of the application software can be done by using YaST for SUSE Linux Enterprise Server 11. To better illustrate the universality of LAMP on both SUSE Linux Enterprise Server and RHEL, the command-line interface (CLI) is used for these instructions, with sample commands for both SUSE Linux Enterprise Server and RHEL.

Install LAMP on SUSE Linux Enterprise Server

First, ensure that SUSE Linux Enterprise Server has a pattern (a collective group of related packages) for a LAMP server. Issue `zypper info -t pattern lamp_server` to see the packages that are associated with a LAMP server.

Example 8-28 shows the helpful information that is displayed about LAMP by running the following command:

```
zypper info -t pattern lamp_server
```

Example 8-28 LAMP pattern output from zypper

```
Insudb2:~ # zypper info -t pattern lamp_server
Loading repository data...
Reading installed packages...
```

```
Information for pattern lamp_server:
```

```
Repository: SLES_DVD
Name: lamp_server
Version: 11-38.44.33
Arch: s390x
Vendor: SUSE LINUX Products GmbH, Nuernberg, Germany
Installed: No
Summary: Web and LAMP Server
Description:
Software to set up a Web server that is able to serve static, dynamic, and inter
active content (like a Web shop). This includes Apache HTTP Server, the database
management system MySQL, and scripting languages such as PHP, Python, Ruby on R
ails, or Perl.
Contents:
```

```
S | Name                | Type    | Dependency
--+-----+-----+-----
  | mysql                | package |
```

```

| apache2-mod_php5      | package |
| apache2-mod_python   | package |
| apache2-prefork      | package |
| libapr-util          | package |
| libapr1              | package |
| apache2               | package |
| apache2-doc          | package |
| apache2-example-pages | package |
1nsudb2:~ #

```

Note: The `lamp_server` pattern includes the Apache and MySQL components, but is missing the PHP component. That is because the “P” in “LAMP”, confusingly, stands for “Ruby,” which is often used as the server-side dynamic web page engine.

Install the packages for Apache and MySQL by issuing the following command:

```
zypper install -t pattern lamp_server
```

The **zypper** command reports which packages are expected to be installed, then prompts for confirmation to continue. Press **y** and **Enter** to install the packages.

Install the remaining PHP packages by issuing the following command:

```
zypper install apache2-mod_php53 php53-mysql
```

Install LAMP on Red Hat Enterprise Linux

Although RHEL has a similar mechanism for representing collections of packages as groups as SUSE Linux Enterprise Server does as patterns, RHEL does not have a group for LAMP packages. Therefore, installing the LAMP packages involves specifying four different groups:

```
yum groupinstall "Web Server" "MySQL Database server" "PHP Support"
```

Yum reports which packages are expected to be installed, then prompts for confirmation to continue. Press **y** and **Enter** to install the packages.

In addition to the packages selected by the indicated groups, another package must be installed on the RHEL server:

```
yum install php-mysql
```

8.6.3 Starting and testing LAMP components

Before migrating the MediaWiki software, configure and test Apache, PHP, and MySQL on the target system to ensure that they are working. This process reduces the number of variables to debug if something goes wrong.

The Apache and MySQL configurations in this example scenario are simple, whereas your configuration might be more complex. Migrating the Apache and MySQL configurations can be a more complex process. This example presumes that MediaWiki is the only application configured for Apache and that no other data exists in the MySQL database than what is used by MediaWiki.

Confirm that the version of Apache is what is expected. A common method of displaying the version is by running the **apachectl -v** command, which is the same for SUSE Linux Enterprise Server and RHEL.

Example 8-29 shows the version of apache2 as displayed by issuing this command in SUSE Linux Enterprise Server:

```
apachectl -v
```

Example 8-29 Output of apachectl -v

```
lmsudb2:~ # apache2ctl -v
Server version: Apache/2.2.12 (Linux/SUSE)
Server built: March 27 2013 18:57:40
```

Historically, it was common to have the installed services started automatically when the package was installed. Today, it is more common that the installer help ensure that malicious software is not started 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 SUSE Linux Enterprise Server

Set the apache2 service to automatically start each time that the server is booted, then manually start the service by using the following commands:

```
chkconfig apache2 on
service apache2 start
```

Example 8-30 shows the expected output from the commands that start the apache2 web service.

Example 8-30 Starting apache2 web service

```
lmsudb2:~ # chkconfig apache2 on
lmsudb2:~ # service apache2 start
Starting httpd2 (prefork)                               done
```

Apache services on Red Hat Enterprise Linux

The commands on RHEL are identical to SUSE Linux Enterprise Server, but the name of the service is httpd rather than apache2, as shown in the following commands:

```
chkconfig httpd on
service httpd start
```

Verifying that web server is running

With the web service started, use a web browser to verify that the web server is working as expected, as shown in Figure 8-1. Start a web browser and point it to the IP address of the Linux server. In this example, the URL is `http://9.12.7.90`.

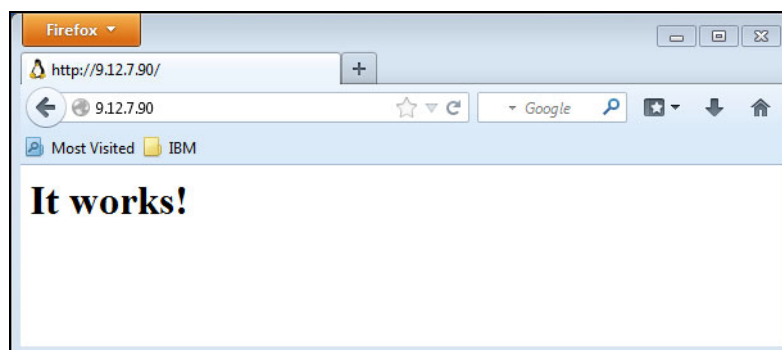


Figure 8-1 Successful test of Apache installation

Verifying that PHP is working

Before a test can be conceived and run for PHP, the location of the DocumentRoot directory of the Apache server must be determined.

In SUSE Linux Enterprise Server, the default location is `/srv/www/htdocs`. However, a non-default location might have been configured. The document root directory can be determined by running the commands that are shown in Example 8-31.

Example 8-31 Finding the DocumentRoot on SUSE Linux Enterprise Server

```
root@lnsadb2:~ # grep 'DocumentRoot "' /etc/apache2/default-server.conf
DocumentRoot "/srv/www/htdocs"
```

Under RHEL, the default location is `/var/www/html`, but the definitive value is revealed by the commands shown in Example 8-32.

Example 8-32 Finding the DocumentRoot on RHEL

```
[root@zs4p01-r1 ~] 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 8-33, and place the script file in the appropriate DocumentRoot directory.

Example 8-33 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 the following URL as an example:

```
http://9.12.7.90/phpinfo.php
```

Figure 8-2 shows the expected PHP information that is generated in the browser by the PHP script running on SUSE Linux Enterprise Server 11 SP3.

PHP Version 5.3.17	
System	Linux lnsub2.itso.ibm.com. 3.0.76-0.11-default #1 SMP Fri Jun 14 08:21:43 UTC 2013 (ccab990) s390x
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/conf.d

Figure 8-2 PHP configuration information generated by phpinfo.php

Start MySQL services on SUSE Linux Enterprise Server

Set the MySQL database service to automatically start each time that the server is booted, then manually start the service using these commands:

```
chkconfig mysql on
service mysql start
```

Start MySQL services on Red Hat Enterprise Linux

As before with Apache, the MySQL database service is named slightly differently in RHEL than it is in SUSE Linux Enterprise Server. To set MySQL to start each time that the server is booted, and to manually start the service, issue the following commands:

```
chkconfig mysqld on
service mysqld start
```

Note: SUSE Linux Enterprise Server 11 SP3 and RHEL 6.4 both use standard SysVinit commands. SUSE Linux Enterprise Server 12 and RHEL 7 are both slated to replace SysVinit with systemd for the management of services.

Verifying that MySQL is working

MySQL must be configured and working properly before MediaWiki can even be installed. Complete these steps to verify that MySQL is working:

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 good idea. Example 8-34 shows sample commands.

Later, when migrating from the x86 server, you will likely copy the `my.cnf` file from the x86 server to LinuxONE. For now, the example `my.cnf` configuration file is sufficient to test the functions of the system before migrating.

Example 8-34 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. Preferred 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 administrative user. Remember this password because it is required during a few more steps of the process before migrating the MediaWiki application from the x86 server. This can be the same password as the MySQL database that will later be migrated, but does that have to be. Use the command shown in Example 8-35 to set the password.

Example 8-35 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 require providing a password. General administrative functions require the root password, whereas commands that involve MediaWiki 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 contains special characters like 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 might make reproducing the password later an inconvenient mystery.

Test the MySQL capabilities by running this sample command:

```
mysql -u root -p -e "show tables" mysql
```

The preceding command prompts you for the root password that you set in the previous steps with the `mysqladmin` command. The sample output displayed in Example 8-36 shows the list of tables that are contained in the `mysql` database, proving that you have properly set the password.

Example 8-36 Output from the “show tables” mysql command after providing password

```
lnsudb2:~ # mysql -u root -p -e "show tables" mysql
Enter password:
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| func            |
| help_category   |
| help_keyword    |
| help_relation   |
| help_topic      |
| host            |
| proc            |
| procs_priv      |
| 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, you can install the MediaWiki software. If the MySQL administrative password was set up incorrectly, an error message similar to Example 8-37 is displayed.

Example 8-37 Bad password supplied to MySQL

```
lnsudb2:~ # mysql -u root -p -e "show tables" mysql
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password:
ES)
```

To correct this problem, run the `mysqladmin` command again as shown in Example 8-35 on page 157, 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, you must reinstall MySQL.

Note: Another migration option for MySQL is to run `mysqldump` on the x86 server, transfer the resulting MySQL dump files to the LinuxONE server, and restore the database with `mysqlimport`. This is a widely used practice, but your environment might dictate a different approach. 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 LinuxONE server, the iSCSI disks can now be migrated from the x86 server.

8.6.4 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 is the case for this MediaWiki migration example.

In this example, the MySQL database and the Apache DocumentRoot are each contained on separate iSCSI disk partitions. The file systems on those partitions will be unmounted from the `zs4p01-s1` host (an x86 system), then mounted on the `Insudb2` host running on LinuxONE.

Note: When dealing with remote disk storage, 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 is used persistently when the host is rebooted. The only persistent identification is by-path.

Prepare Linux Guests on LinuxONE for iSCSI

Before making any changes on the x86 host using the iSCSI LUNs, set up the minimum iSCSI software on the Linux guest running on LinuxONE:

1. Connect to the LinuxONE guest called `1nsudb2` by using Secure Shell (SSH).
2. Ensure that the iSCSI initiator software is installed on LinuxONE guest `Insudb2`:
 - For SUSE, use `zypper install open-iscsi`
 - For RHEL, use `yum install iscsi-initiator-utils`
3. Stop the Apache and MySQL services running on `1nsudb2`. This guest ran Apache and MySQL for the earlier tests.
 - For SUSE, use these commands:

```
service apache2 stop
service mysql stop
```
 - For RHEL, use these commands:

```
service httpd stop
service mysqld stop
```

4. Move the content of the `/srv/www` and `/var/lib/mysql` directories out of the way so that the file systems of the iSCSI remote LUNs can be mounted in their places. However, keep the contents of these directories available as a backup. The default DocumentRoot for RHEL is `/var/lib/html/`. Use the proper directory for your circumstances.

```
mv /srv/www /srv/www.orig
mv /var/lib/mysql /var/lib/mysql.orig
mkdir -p /srv/www /var/lib/mysql
```

Prepare x86 system for migration

Complete these steps to prepare your x86 system for migration:

1. Connect to the `zs4p01-s1` (x86) host by using SSH.
2. Display the mount table of `zs4p01-s1`, 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 in `/dev/sdd1` and the `/srv/www` directory is mounted in `/dev/sdc1`. Example 8-38 shows a similar method of displaying the mounted file systems by using `df -h`.

Example 8-38 Mounted disk partitions on `zs4p01-s1`

```
zs4p01-s1:/var/lib # df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       491G   29G  462G   6% /
udev            2.0G  116K   2.0G   1% /dev
tmpfs           2.0G   72K   2.0G   1% /dev/shm
/dev/loop0      3.2G   3.2G    0 100% /srv/ISO
/dev/sdc1       1018M  199M   820M  20% /srv/www
/dev/sdd1       4.0G   33M   3.9G   1% /var/lib/mysql
```

3. Take note of which remote LUNs are currently being mounted on host `zs4p01-s1`:

```
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-9.12.7.97:3260-iscsi-iqn.2014-04.ibm.itso:LUN2-lun-2-part1
```

where:

- The IP address of the remote iSCSI target = `9.12.7.97`
- The iSCSI Target Identifier = `iscsi-iqn.2014-04.ibm.itso`
- The unique iSCSI LUN for the Apache `www` partition = `LUN2`

For the MySQL data directory, the IP address of the remote iSCSI target and the iSCSI Target Identifier are the same as those of the Apache `www` disk. The only difference is that the unique iSCSI LUN for the MySQL data partition is equal to `LUN3`.

Example 8-39 shows the by-path allocations for this example, with LUN2 being referenced as sdb1 and LUN3 being referenced as sdc1.

Example 8-39 Output showing the by-path assignments of remote iSCSI disks

```
zs4p01-s1:~ 1 /dv/disk/by-path/
total 0
lrwxrwxrwx 1 root root 9 May 8 09:14 ip-9.12.7.97:3260-iscsi-iqn.2014-04.ibm.itso:LUN2-lun-2 ->
../../sdc
lrwxrwxrwx 1 root root 10 May 8 09:14
ip-9.12.7.97:3260-iscsi-iqn.2014-04.ibm.itso:LUN2-lun-2-part1 ->../../sdc1
rwxrwxrwx 1 root root 9 May 8 09:14 ip-9.12.7.97:3260-iscsi-iqn.2014-04.ibm.itso:LUN3-lun-3 ->
../../sdd
lrwxrwxrwx 1 root root 10 May 8 09:14
ip-9.12.7.97:3260-iscsi-iqn.2014-04.ibm.itso:LUN3-lun-3-part1 ->../../sdd1
```

LUN2 is a symbolic link to /dev/sdc1 and LUN3 is a symbolic link to /dev/sdd1. These directories are the partitions from the LUNs that were identified in the previous step, and hence are the file systems to be unmounted.

Note: Pay close attention to these details. Selecting the wrong disk can result in unmounting the wrong file system, which can have disastrous consequences.

4. Stop Apache and MySQL running on zs4p01-s1 (x86) host:

- For SUSE, use these commands:

```
service apache2 stop
service mysql stop
```

- For RHEL, use these commands:

```
service httpd stop
service mysqld stop
```

5. Unmount the disks that contain the file systems that will be moved to lnsudb2:

```
umount /srv/www
umount /var/lib/mysql
```

6. For completeness, log out of the two LUNs of the iSCSI target connected from zs4p01-s1:

```
iscsiadm --mode=node --portal=9.2.7.97 --targetname=iqn.2014-04.ibm.itso:LUN2
--logout
iscsiadm --mode=node --portal=9.2.7.97 --targetname=iqn.2014-04.ibm.itso:LUN3
--logout
```

7. Remove the records that refer to the /srv/www and /var/lib/mysql from the zs4p01-s1 host's /etc/fstab. You might want to retain the information before you remove it. It is likely that you will use the same information on the new host, lnsudb2.

Move iSCSI disks to LinuxONE

Complete these steps to move your iSCSI disks to LinuxONE:

1. Use an SSH to connect again to the console of guest LNSUDB2. No more work will be done using the console on the x86 host zs4p01-s1.
2. Discover the remote disk services that are running on the remote iSCSI target:

```
iscsiadm --mode=discovery --type=sendtargets --portal=9.12.7.97
```

Example 8-40 shows three of the LUNs that are available from the iSCSI target. The example uses only two of the LUNs.

Example 8-40 The LUNs discovered from the iSCSI target

```
lnsudb2:~ # iscsiadm --mode discovery --type sendtargets --portal 9.12.7.97
9.12.7.97:3260,1 iqn.2014-04.ibm.itso:LUN2
9.12.7.97:3260,1 iqn.2014-04.ibm.itso:LUN1
9.12.7.97:3260,1 iqn.2014-04.ibm.itso:LUN3
```

The output in Example 8-40 is similar to the information seen on zx4p01-s1, 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 lnsudb2 are LUN2 (which contains the Apache www file system) and LUN3 (containing the MySQL database files). Although LUN1 is also listed, LUN1 is not used in this exercise:

```
9.12.7.97:3260,1 iqn.2014-04.ibm.itso:LUN2
9.12.7.97:3260,1 iqn.2014-04.ibm.itso:LUN3
```

3. Log in to the remote iSCSI disk, specifying the LUN (LUN2) for the Apache www disk:

```
iscsiadm --mode=node --portal=9.12.7.97 --targetname=iqn.2014-04.ibm.itso:LUN2
--login
```

The output that is shown in Example 8-41 represents a successful login.

Example 8-41 Successful login to iSCSI target's LUN2

```
lnsudb2:~ # iscsiadm --mode=node --portal=9.12.7.97 --targetname=iqn.2014-04.ibm
.itso:LUN2 --login
Logging in to [iface: default, target: iqn.2014-04.ibm.itso:LUN2, portal: 9.12.7
.97,3260] (multiple)
Login to [iface: default, target: iqn.2014-04.ibm.itso:LUN2, portal: 9.12.7.97,3
260] successful.
```

In your environment, the iSCSI target will probably require more sophisticated authentication for login. In this simple example, the iSCSI target requires no credentials of any kind.

4. Repeat the login, this time specifying LUN3, which contains the MySQL data partition:

```
iscsiadm --mode=node --portal=9.12.7.97 --targetname=iqn.2014-04.ibm.itso:LUN3
--login
```

Note: The `--login` subcommand command on SUSE Linux Enterprise Server causes the iSCSI initiator to connect *only* to the specified LUN. Therefore, only the specified LUN is viewable. However, with RHEL, the `--login` subcommand to the iSCSI target allows 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 means that only one login step is needed. One challenge of it is that after the login is accomplished, all the iSCSI LUNs are given `/dev/sdX` assignments regardless of whether they are all wanted.

5. See that the remote LUNs are now connected to host lnsudb1:

```
ls -l /dev/disk/by-path/
```


Example 8-42 shows the partition sda1 being mapped to iSCSI LUN2, and partition sdb1 being mapped to LUN3.

Example 8-42 iSCSI LUNs mapped to disk devices after login

```
lnsudb2:- # 1 /dev/disk/by-path/
total 0
drwxr-xr-x 2 root root 200 May 6 09:16 ./
drwxr-xr-x 5 root root 100 Apr 18 11:41 ../
lrwxrwxrwx 1 root root 11 Apr 18 11:41 ccw-0.0.0201 -> .././dasda
lrwxrwxrwx 1 root root 12 Apr 18 11:41 ccw-0.0.0201-part1 -> .././dasda1
lrwxrwxrwx 1 root root 11 Apr 18 11:41 ccw-0.0.0202 -> .././dasdb
lrwxrwxrwx 1 root root 12 Apr 18 11:41 ccw-0.0.0202-part1 -> .././dasdb1
lrwxrwxrwx 1 root root 9 May 6 09:12 ip-9.12.797:3260-iscsi-iqn.2014-04.1bm.
itso:LUN2-lun-2-> .././sda
lrwxrwxrwx 1 root root 10 May 6 09:12 ip-9.12.7.97:3260-iscsi-iqn.2014-04.1bm.
itso:LUN2-lun-2-> part1 .././sda1
lrwxrwxrwx 1 root root 9 May 6 09:16 ip-9.12.7.97:3260-iscsi-iqn.2014-04.1bm.
itso:LUN3-lun-3 -> .././sdb
lrwxrwxrwx 1 root root 10 9 May 6 09:16 ip-9.12.7.97:3260-iscsi-iqn.2014-04.1bm.
itso:LUN3-lun-3-part1 -> .././sdb1
```

6. 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 8-43 shows a snippet from `/etc/fstab`, with the two new file system entries.

Example 8-43 New /etc/fstab containing the new MySQL and www disks on lnsudb2

```
# external iSCSI disk LUN2 for www filesystem
/dev/disk/by-path/ip-9.12.7.97:3260-iscsi-iqn.2014-04.1bm.itso:LUN2-lun-2-part1
/srv/www          xfs          nofail          1 2
# external iSCSI disk LUN3 for MySQL filesystem
/dev/disk/by-path/ip-9.12.7.97:3260-iscsi-iqn.2014-04.1bm.itso:LUN3-lun-3-part1
/var/lib/mysql    xfs          nofail          1 2
```

7. Mount the remote iSCSI disks in their proper places by using the following commands:

```
mount /var/lib/mysql
mount /srv/www
```

The file systems from the remote iSCSI disks are now mounted and available on the new LNSUDB2 host.

Complete migration of services

The remaining tasks are critical and the most specialized. Copy the Apache and MySQL configuration files from the x86 host to the LinuxONE guest, then adapt them. Notice that the configuration of MediaWiki requires no special steps because it was all moved when the www partition was moved. In some cases, the configurations are simple enough to allow a basic copy of the files. Other circumstances can be more complex, which require rewriting the configuration files. However, this effort is acceptable, given that the data is migrated so effortlessly by using iSCSI.

For this example, a simple copy of the configuration files is all that is necessary. To do this, complete the following steps:

1. Start an SSH console on LNSUDB2 (LinuxONE).
2. Synchronize the Apache configuration files to LNSUDB2 from zs4p01-s1. Use the method that makes the most sense for your environment, following this example:

```
rsync -qa zs4p01-s1:/etc/apache2/* /etc/apache2/  
rsync -qa zs4p01-s1:/etc/my.cnf /etc/
```

3. Start the Apache and MySQL services on LNSUDB2:

- For SUSE, use these commands:

```
service apache2 start  
service mysql start
```

- For RHEL, use these commands:

```
service httpd start  
service mysqld start
```

4. Ensure that Apache and MySQL start each time that the server is booted:

- For SUSE, use these commands:

```
chkconfig apache2 on  
chkconfig mysql on
```

- For RHEL, use these commands:

```
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. Open the MediaWiki URL using a browser. The web page shown in Figure 8-3 represents a successful installation of MediaWiki.

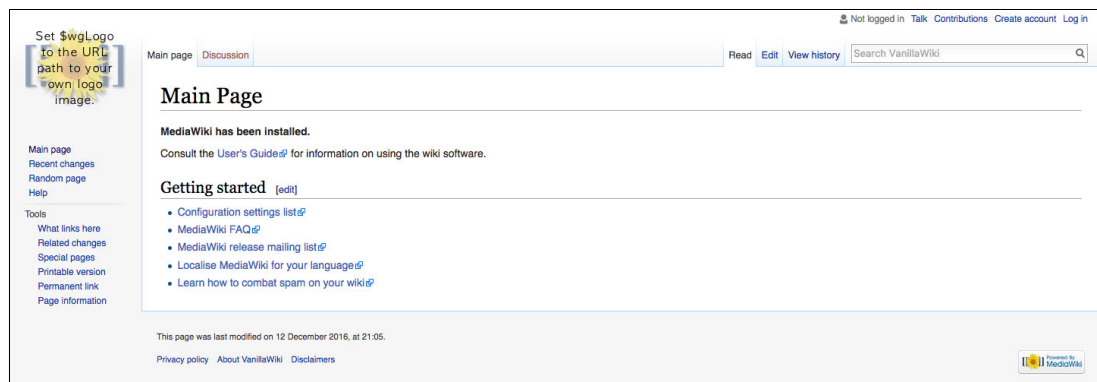


Figure 8-3 A successful migration of MediaWiki

The page shown in Figure 8-3 originally comes from www.wikipedia.org. A small portion of a backup file for the Wikipedia site was used for this example. The backup file was retrieved from <http://dumps.wikimedia.org> and restored in the test environment in our labs for demonstration purposes only. The content and copyrights of the page shown are property of Wikimedia Foundation, Inc.® and are used in accordance with the Terms of Use supplied by www.wikipedia.org.

8.7 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 (LDAP).

LDAP is widely used throughout the industry for directory services as an open standard running over an IP network. Although several commercial LDAP products are 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 LinuxONE from both RHEL and SUSE. LDAP is a perfect workload for LinuxONE, due to the centrality of LinuxONE among many other systems and services, its fast I/O, and its low CPU and memory usage. And OpenLDAP is open source. Migrating OpenLDAP to LinuxONE is straightforward.

We installed a LAMP server with MediaWiki, and iSCSI external storage was used to facilitate the migration in 8.6, “Deploying MediaWiki and MySQL” on page 152. In this example, the LDAP database on an x86 server is exported, the database is transferred to a Linux guest running on LinuxONE, and the data is imported into the LDAP service.

8.7.1 Analysis and planning

As with the MediaWiki example described in 8.6, “Deploying MediaWiki and MySQL” on page 152, it is important that you follow Chapter 4, “Migration process” on page 43, and Chapter 5, “Migration analysis” on page 55. Perform this planning and analysis before any migration activity. The checklists help identify the many considerations that should be taken into account to help prevent problems during migration.

This example assumes that the Linux 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 Linux Enterprise Server 11 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 LinuxONE. It is presumed that an adequate RPM repository installation source is already set up and available for the installation of the application software.

The x86 server is called zs4p01-r1 and is running RHEL 6.4. For this example, this is the current OpenLDAP server that provides directory services for the hypothetical organization. This server has a rudimentary (small) LDAP directory already configured.

Although there is much to consider when setting up an enterprise directory service, a simple OpenLDAP scenario is covered here. More extensive documentation can be found at the following site:

<http://www.openldap.org>

This example is a stand-alone server with a local, non-replicated directory service. Nevertheless, migrating an existing OpenLDAP installation on x86 to LinuxONE should be straightforward.

8.7.2 Installing LDAP software

The OpenLDAP server is a simple application, consisting of a single package. Therefore, installing the software is relatively easy. The software must first be installed on the LinuxONE guest before the other migration steps. If you are going to install OpenLDAP on SUSE Linux Enterprise Server, run the following command to install the package:

```
zypper install openldap2
```

To install OpenLDAP on RHEL, run the following command:

```
yum install openldap-servers
```

8.7.3 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 serve as the basis of the example service that is configured here.

The initial configuration of OpenLDAP on SUSE Linux Enterprise Server running on LinuxONE is accomplished using YaST, whereas configuration on Red Hat is done by manually modifying configuration files and running commands.

Before migrating the LDAP database to LinuxONE, 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 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 Linux Enterprise 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. Perform the following steps:

1. From a command prompt, start YaST, calling specifically the `ldap-server` module:

```
yast2 ldap-server
```

Figure 8-4 shows the first panel.

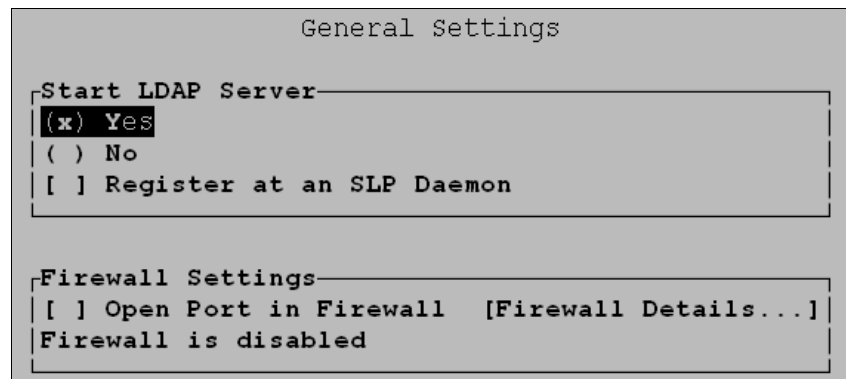


Figure 8-4 `yast2 ldap-server` module

Select **Yes** to start the LDAP server automatically. Be certain to open a port in the firewall. In this example, because the firewall is disabled, we did not have the option to select **Open Port in Firewall**.

Press **F10** to go to the next panel.

2. Select **Stand-alone server** as the server type, as shown in Figure 8-5.

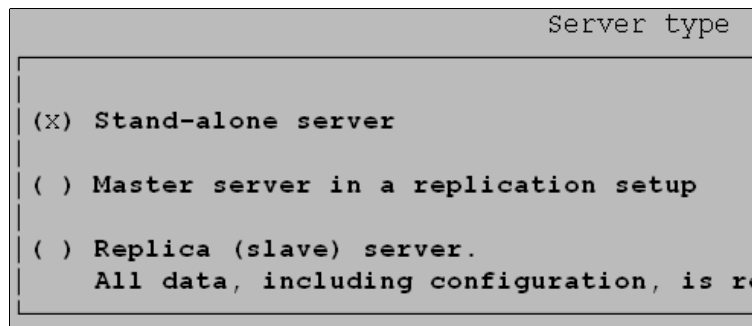


Figure 8-5 Stand-alone server type of the LDAP server

Press **F10** to go the next panel.

3. Select the proper security settings for OpenLDAP, as shown in Figure 8-6.

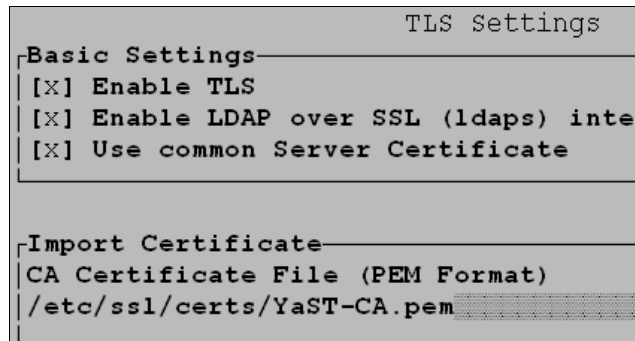


Figure 8-6 TLS and SSL certificate settings for OpenLDAP

Using a proper SSL certificate is preferred, but not necessary for this demonstration. This example uses the self-signed system certificates that were generated when SUSE Linux Enterprise Server was installed. More importantly, using SSL for LDAP (also known as Lightweight Directory Access Protocol Over Secure Socket Links (LDAPS)) is essential. Without LDAPS, passwords and other sensitive data are exchanged with the LDAP server in plaintext. This method makes the system vulnerable.

Press **F10** to go to the next panel.

4. Figure 8-7 shows the Basic Database Settings panel, which includes fields for setting an administrative password for LDAP.

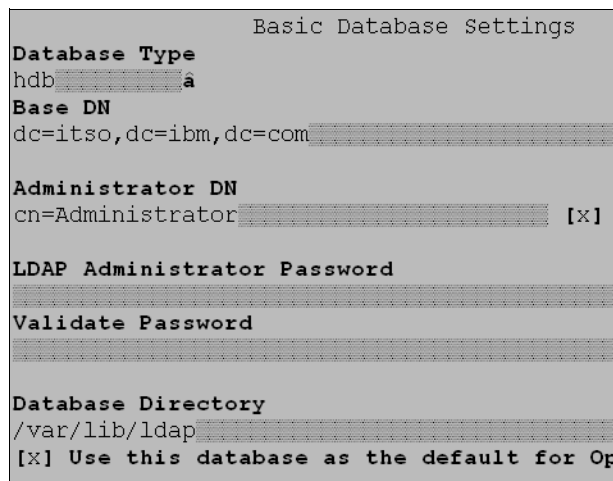


Figure 8-7 Basic database settings for OpenLDAP configuration

In a production environment, proper distinguished name (DN) data must be entered, but for this demonstration it is adequate to use the sample values supplied by YaST. What is most important here is providing an administrator password. This password should not be the same as the system's root password. All other preferred practices for creating an administrative password should likewise be employed. For this demonstration, the password `ldapadmin` is used.

Press **F10** to go to the next panel.

5. The configuration summary is displayed, as shown in Figure 8-8.

```
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 8-8 OpenLDAP configuration summary

With all the configuration information sufficiently gathered, the YaST configuration steps can be completed by pressing **F10**. The configuration files are written, and the `slapd` daemon is started. The running daemon process can be seen in Example 8-44. The `-F /etc/openldap/slapd.d` argument indicates that the service is configured by using the `cn=config` feature format.

Example 8-44 `slapd` daemon shown running using the `cn=config` method

```
lnsadb2:- # 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 allows 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:
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 8-45 into the file.

Example 8-45 `/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 8-45 using the following command:

```
ldapadd -Y EXTERNAL -H ldapi:/// -f /tmp/config.itso.ibm.com.ldif
```

Now the basic configuration of OpenLDAP allows a migration of the database.

8.7.4 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 LinuxONE. To do this, complete the following steps:

1. Connect to the x86 host, `zs4p01-r1`, using SSH. This example is 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*, which 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`. Because the daemon is being migrated to another server, it might not be necessary to restart it.

```
service slapd start
```

5. Transfer the database file to the Linux guest, `LNSUDB2`, running on LinuxONE. 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/
```

The server with the IP address `9.12.7.90` is `LNSUDB2` and is the Linux guest on LinuxONE. 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 LinuxONE is completed.

8.7.5 Import OpenLDAP data to LinuxONE

In the previous section, the OpenLDAP database export file was transferred to `lnsldb2`, the Linux guest running on LinuxONE. All that is required now is to import the data and start the OpenLDAP daemon:

1. Reconnect to the Linux guest, `lnsldb2`, 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 8.7.3, “Configuring the OpenLDAP service” on page 166, the `itso.ibm.com` domain exists in the new OpenLDAP database, making it 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 shown in Example 8-46. Any value other than 100% means that something went wrong and the import of the data was not successful.

Example 8-46 Import of OpenLDAP data is 100% successful

```
lnsldb2:- # slapadd -F /etc/openldap/slapd.d -b 'dc=itso,dc=ibm,dc=c  
om' -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. After the database has been successfully imported, OpenLDAP can be started again, ready to receive queries:

```
service slapd start
```

8.7.6 Verify 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. 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 8-47 shows the results of the `ldapsearch` query.

Example 8-47 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
```

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 8-48 shows that a different client, zs4p01-s1, queries the LDAP directory running on lnsudb2 (9.12.7.90).

Example 8-48 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 8-48 indicates a successful migration of an OpenLDAP service from Linux on x86 to LinuxONE. Not only that, but the service has been easily migrated from a system running RHEL to one running SUSE Linux Enterprise Server. OpenLDAP, Linux, and LinuxONE are all happy regardless of the distribution, and the migration of OpenLDAP is unhampered regardless of the distribution.

8.8 Deploying central log server

As you saw in “Logging and recording events” on page 100, 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, use the default log daemon from SUSE, `syslog-ng` (version 2.09). It can be easily installed on RHEL by using Extra Packages for Enterprise Linux (EPEL6).

8.8.1 Analysis and planning

Use the Logical Volume Manager (LVM) to create a logical volume for log files because log files tend to grow fast. With different hosts writing logs to a centralized log server at the same time, log files can fill your disk even faster. Therefore, leave some space available in a volume group that can be used during an emergency.

8.8.2 Initial configuration

The default path for the `syslog-ng` configuration file is `/etc/syslog-ng/syslog-ng.conf`. It is made up of key words that define the message route and global options. You can see all the available global options by issuing the following command:

```
man syslog-ng.conf
```

Global options

You can specify several global options in the options statement of `syslog-ng`. You can define how the host name of the client appears in the log files, enable or disable DNS cache, use the ownership of the files and some other features that you use depending on the size or specific requirements of your environment.

Source options

The source keyword is used to add source drivers or define your own sources. For example, the syntax to enable `syslog-ng` and define a UDP or TCP connection to listen for remote logs is the following:

```
source s_net { tcp((ip(127.0.0.1) port(40000) max-connections 5000)); udp (); };
```

This entry in the configuration files defines a source called `s_net` that listens on localhost to port 40000 using TCP and listen port 514 (default for `syslog-ng`) using UDP on all interfaces. Also, it is limiting the maximum number of connections of the TCP port. For more information about parameters that you can define, see the `syslog-ng.conf` manual.

Filter options

Filters allow you to set different destinations depending on certain key words. The syntax for the filter statement is as follows:

```
filter <identifier> { expression; };
```

where `<identifier>` is the name that you give your filter and `<expression>` contains the function, and Boolean operators (and, or, not). Example 8-49 demonstrates this filter.

Example 8-49 Filter examples

```
filter f_iptables { facility(kern) and match("IN=") and match("OUT="); };
filter f_messages { not facility(news, mail) and not filter(f_iptables); };
filter f_warn { level(warn, err, crit) and not filter(f_iptables); };
```

Destination options

Destination options define the files to which log messages are sent. The `syslog-ng.conf` default file covers the destination for local files, but what about the incoming messages? By default, the remote log messages are written to the default local files depending on what kind of message it received. If it is an authentication message, the log is written to the file `/var/log/auth` with the appended information defined in the global options such as host name, date, and time. Example 8-50 uses files as a destination, but you can use databases or even another remote server, depending on how you configure your server.

Example 8-50 Destination example

```
#
# All messages except iptables and the facilities news and mail:
#
destination messages { file("/var/log/messages"); };
log { source(src); filter(f_messages); destination(messages); };
#
# Firewall (iptables) messages in one file:
#
destination firewall { file("/var/log/firewall"); };
log { source(src); filter(f_iptables); destination(firewall); };
#
# Warnings (except iptables) in one file:
```

```
#
destination warn { file("/var/log/warn" fsync(yes)); };
log { source(src); filter(f_warn); destination(warn); };
```

As you can see in Example 8-50 on page 173, the log statement requires a source, filter, and destination.

8.8.3 Server configuration

A complex server configuration is out of the scope of this book. The default configuration file of `syslog-ng` comes with definitions that split your files depending on the facility that will be logged. Figure 8-9 shows a centralized `syslog-ng` server receiving log copies from `syslog-ng` clients.

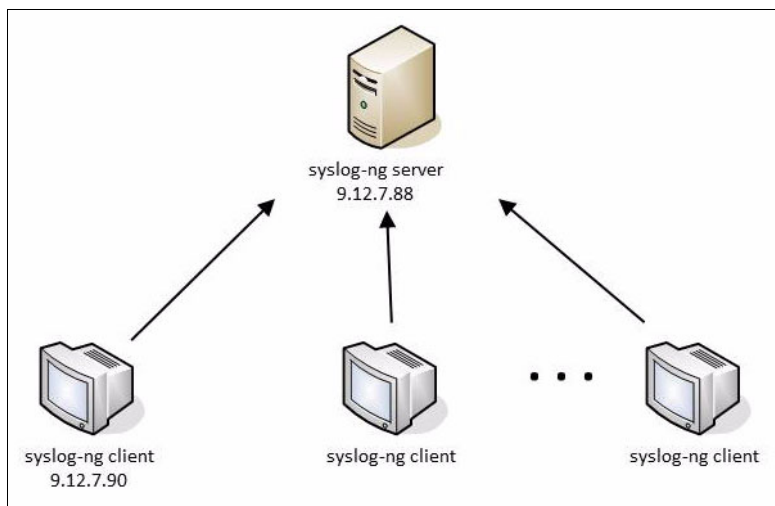


Figure 8-9 A centralized `syslog-ng` server and some `syslog-ng` clients

The global options for the test environment are shown in Example 8-51.

Example 8-51 Global options for `syslog-ng` server

```
options {
sync(0); # The number of lines buffered before written to file
perm(0640); # Permission value for created files.
keep_hostname(yes); # Keeps the hostname from the origin.
};
```

To enable the listener, define the source (`src`) to use the `udp` statement, as shown in Example 8-52.

Example 8-52 Source example for `syslog-ng` server

```
source src {
#
# include internal syslog-ng messages
# note: the internal() source is required!
#
internal();

#
```

```
# the default log socket for local logging:
#
unix-dgram("/dev/log");

#
# uncomment to process log messages from network:
#
udp(ip("0.0.0.0") port(514));
};
```

Restart the syslog service to load the syslog-ng configuration file as shown in Example 8-53.

Example 8-53 Restarting syslog-ng server to update the new configuration

```
syslog-server-1:~ # service syslog restart
Shutting down syslog services done
Starting syslog services done
```

To test the listener, you can use `lsnf` as shown in Example 8-54.

Example 8-54 Testing the listener with the lsof command

```
syslog-server-1:~ # lsof -i UDP:514
COMMAND      PID USER  FD  TYPE  DEVICE  SIZE/OFF  NODE NAME
syslog-ng    60700 root   4u  IPv4  3279255      0t0  UDP *:shell
```

You can also use TCP if you would like, but to start a simple syslog-ng server, you only need the `udp` statement, as shown in Example 8-52.

8.8.4 Client configuration

To configure the syslog-ng client, a destination needs to be created to point to the log server. In addition, a new log statement needs to be added to point the client source to the new destination. Append the lines shown in Example 8-55 to the `syslog-ng.conf` of the client, using your client's IP address.

Example 8-55 Append these lines to your client configuration to create a destination to the log server

```
destination logserver { udp("9.12.7.88" port(514)); };
log { source(src); destination(logserver); };
```

In this example, all logs from the client are also sent to 9.12.7.88 (lnsuwas1) through UDP using port 514. This is a simple configuration, but you can set up filters, new destinations, and sources depending on the requirement of your environment.

Restart your syslog-ng client as shown in Example 8-56.

Example 8-56 Restarting the syslog-ng client to update the configuration

```
syslog-client-1:~ # /etc/init.d/syslog restart
Shutting down syslog services done
Starting syslog services done
```

8.8.5 Testing syslog-ng

From the log server (syslog-server-1), you can use the command `tail -f` to monitor the messages written to the `/var/log/message` file. See Example 8-57.

Example 8-57 Monitoring the syslog-ng server

```
syslog-server-1:~ # tail -f /var/log/messages
May  2 11:45:01 syslog-server-1 syslog-ng[32617]: Configuration reload request
received, reloading configuration;
May  2 11:45:01 syslog-server-1 syslog-ng[32617]: New configuration initialized;
```

To test the client, you can use the `logger` command:

```
Logger "Testing syslog-ng"
```

Example 8-58 shows the results from the log server (syslog-server-1).

Example 8-58 Getting logs from the syslog-ng clients

```
syslog-server-1:~ # tail -f /var/log/messages
May  2 11:45:01 syslog-server-1 syslog-ng[32617]: Configuration reload request
received, reloading configuration;
May  2 11:45:01 syslog-server-1 syslog-ng[32617]: New configuration initialized;
May  2 11:50:39 syslog-client-1 root: Testing syslog-ng
```

For alternative setups, see the official syslog documentation website:

<http://www.balabit.com/support/documentation>

8.8.6 Migrating using syslog-ng

You can use `syslog-ng` 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. With this configuration, 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 `syslog-ng` server/client, check whether `syslog-ng` is installed on the target server. Also, ensure that the former configuration file is compatible with the version available on LinuxONE. 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 *Set up Linux on IBM System z for Production*, SG24-8137.

8.9 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 several similar commercial products are available, Samba is the implementation that is most commonly used in Linux environments to share files and printers. It is available for LinuxONE from both Red Hat and SUSE and allows interoperability between UNIX/Linux servers and Windows/Linux based clients. Samba runs easily on LinuxONE because its hardware has fast I/O that provides high-performance access to applications and files.

Before you deploy Samba, ensure that appropriate analysis and planning has been performed before any migration activity. The checklists that are provided in this book help identify the many areas to take into consideration to help prevent problems during migration.

This example assumes that 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 Linux Enterprise Server 11 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 LinuxONE. It is presumed that an adequate RPM repository installation source is already set up and available for the installation of the application software.

More extensive documentation about Samba can be found at the following site:

<http://www.samba.org>

This example is a stand-alone server with a local, non-replicated directory service. Migrating an existing Samba installation on x86 to LinuxONE should be straightforward.

8.9.1 Installing Samba software

Installing the software is relatively easy:

- ▶ To install Samba and its dependencies packages on SUSE Linux Enterprise Server, issue this command:

```
zypper install samba
```

- ▶ To install Samba and its dependencies packages on Red Hat Enterprise Linux, issue this command:

```
yum install samba
```

8.9.2 Configuring Samba

This section describes how to configure Samba first on SUSE Linux Enterprise Server and RHEL.

Configuring file server on SAMBA on SUSE Linux Enterprise Server using YaST

All of the activities to create a working configuration are facilitated by the Samba server YaST module. By using 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**. Complete the fields that are shown in Example 8-59 with your network information.

Example 8-59 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 can change the settings later in the Samba configuration panels on YaST, as shown in Figure 8-10.

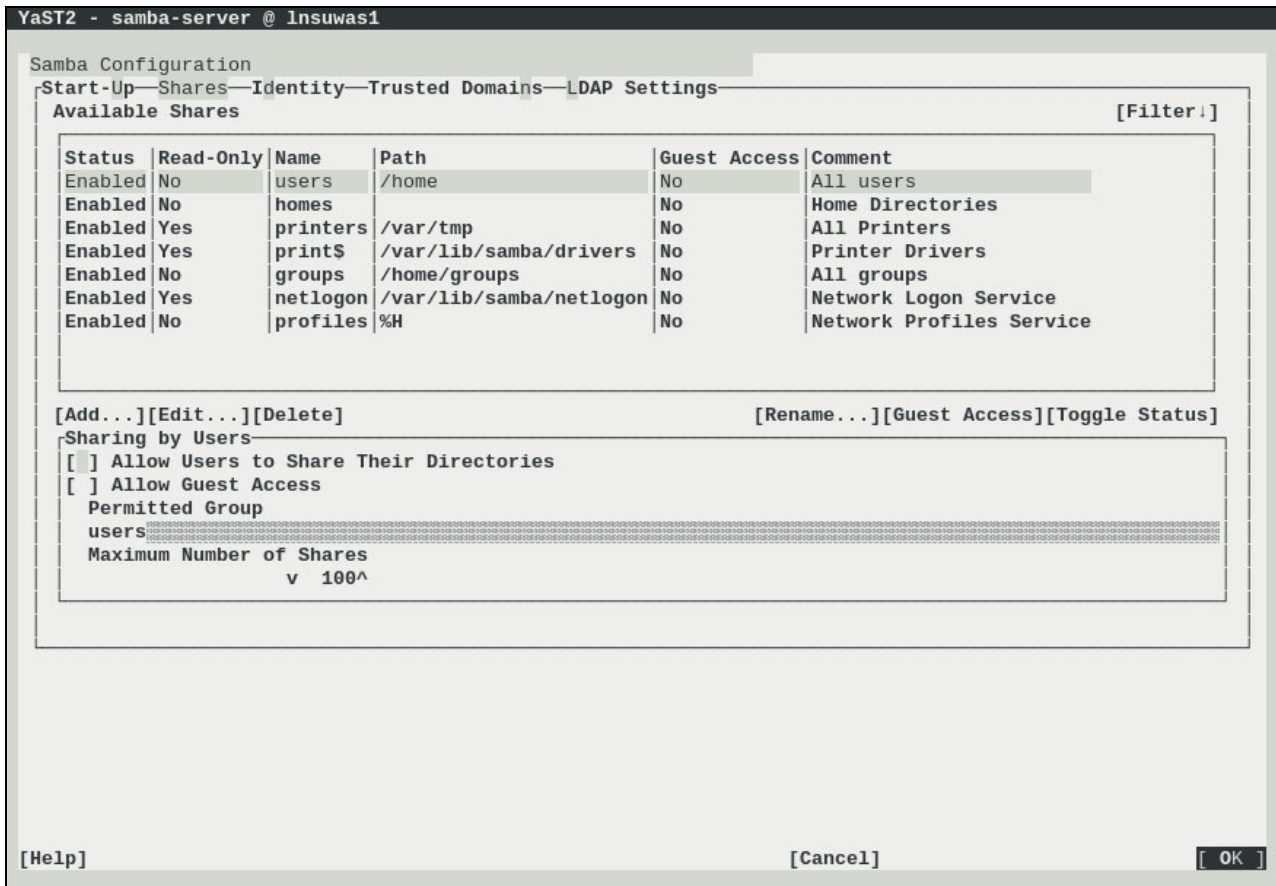


Figure 8-10 Samba Server Configuration tab

Samba shares

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 panel shown in Figure 8-10, select the Shares tab and then select **Add**. Provide the information shown in Example 8-60 and shown on the YaST2 panel in Figure 8-11 on page 179.

Example 8-60 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 8-11 shows the configuration information in the YaST2 panel.

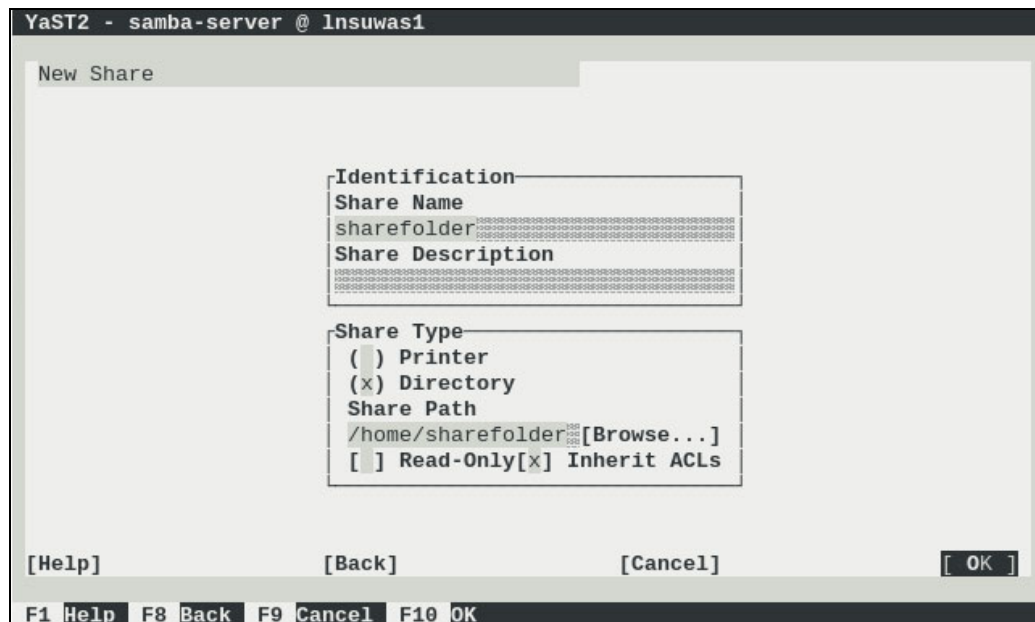


Figure 8-11 Sample of creating new share on Samba using Yast

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 a 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 about how to set up LDAP on Samba, see the Samba official website:

<http://www.samba.org>

Configuration files

You can manually set up configuration files for Samba. The main configuration file on SUSE Linux Enterprise Server is stored in `/etc/samba/smb.conf`, and has two sections:

- ▶ [`global`] for general settings
- ▶ [`share`] to specify specific settings about sharing files and printers

For more information about Samba configuration on SUSE Linux Enterprise Server, see Samba section in the SUSE Linux Enterprise Server 11 Administration Guide:

bit.ly/Zexlvc

Note: RHEL 6 uses the same structure as SUSE Linux Enterprise 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 shown in Example 8-61.

Example 8-61 Stopping and starting the Samba service

To stop and start the service both SLES and RHEL:

Stop the service: `/etc/init.d/smb stop`

Start the service: `/etc/init.d/smb start`



Appendix

This section contains an appendix that provides additional use cases:

- ▶ Appendix A, “Additional use case scenarios” on page 183



A

Additional use case scenarios

The complexity of a migration from Linux on the x86 can change by platform architecture and context of the migration. The Linux operating system is more straightforward and well-known, and makes 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.

This appendix provides additional use case scenarios where a telecommunications company, a healthcare company, and an energy and utilities company all want to migrate from x86 to LinuxONE. It describes the challenges inherent to each industry and their respective migration scenarios.

This appendix includes the following sections:

- ▶ Telecom industry consolidation and cloud
- ▶ Healthcare industry: Mobile and internet solution
- ▶ Energy and utilities industry: SAP Cloud and Automation solution on System z

Telecom industry consolidation and cloud

In this scenario, Fictional Telco Company T1 selects the IBM LinuxONE platform for their Linux operating system consolidation and virtualization. Telco Company T1 wants to build a cloud platform, but they also want to reduce their cost of operation and overall data center footprint. The company's strategy is to improve provisioning time for its business support system (BSS) and operational support system (OSS) to satisfy server requests of its users. In this example, the following technology can be employed:

Fictional Telco Company T1 has this consolidated hardware infrastructure:

- ▶ IBM LinuxONE Emperor or Rockhopper
- ▶ IBM z/VM 6.3
- ▶ Red Hat Enterprise Linux or SUSE Linux Enterprise Servers on the LinuxONE platform
- ▶ IBM ProtecTIER® Gateway TS7680: Deduplication and Virtual Tape Library

Fictional Telco Company T1 uses this cloud application:

- ▶ IBM SmartCloud®:
 - Automation with cloud: IBM Tivoli System Automation
 - Automated provisioning: Tivoli Provisioning Manager
 - Service Lifecycle Management: IBM SmartCloud Control Desk

Fictional Telco Company T1 uses these build monitoring and system management tools:

- ▶ IBM Tivoli OMEGAMON on z/VM and Linux: Information about your Linux instances running as z/VM guests and the Linux workloads reveal how they are performing and affecting z/VM and each other:
 - Compare Linux operations side by side with detailed performance metrics.
 - Data collection from the Performance Toolkit for VM (PTK is a prerequisite) complements data collection by the IBM Tivoli Monitoring for Linux for System z agent.
 - With new Dynamic Workspace Linking, you can easily navigate between Tivoli Enterprise Portal workspaces.
 - View and monitor workloads for virtual machines, groups, response times, and LPAR reporting, as well as view reports about z/VM and Linux usage of resources such as CPU utilization, storage, mini-disks, and TCP/IP.
 - High-level views help executives understand how systems performance influences business and the bottom line.
 - With granular views, IT staffs can more easily track complex problems that span multiple systems and platforms, and share related information.
- ▶ IBM Wave for z/VM v1.1: IBM Wave is a new virtualization management product for z/VM and Linux virtual servers that uses visualization to dramatically automate and simplify administrative and management tasks:
 - Automate, simplify management, and monitor virtual servers and resources, all from a single dashboard.
 - Perform complex virtualization tasks in a fraction of the time compared to manual execution.
 - Provision virtual resources (servers, network, storage) to accelerate the transformation to cloud infrastructure.

- Use advanced z/VM management capabilities such as Live Guest Relocation with a few clicks.
- Delegate responsibility and provide more self-service capabilities to the appropriate teams.

Figure A-1 shows the solution architecture overview for a cloud solution that uses LinuxONE.

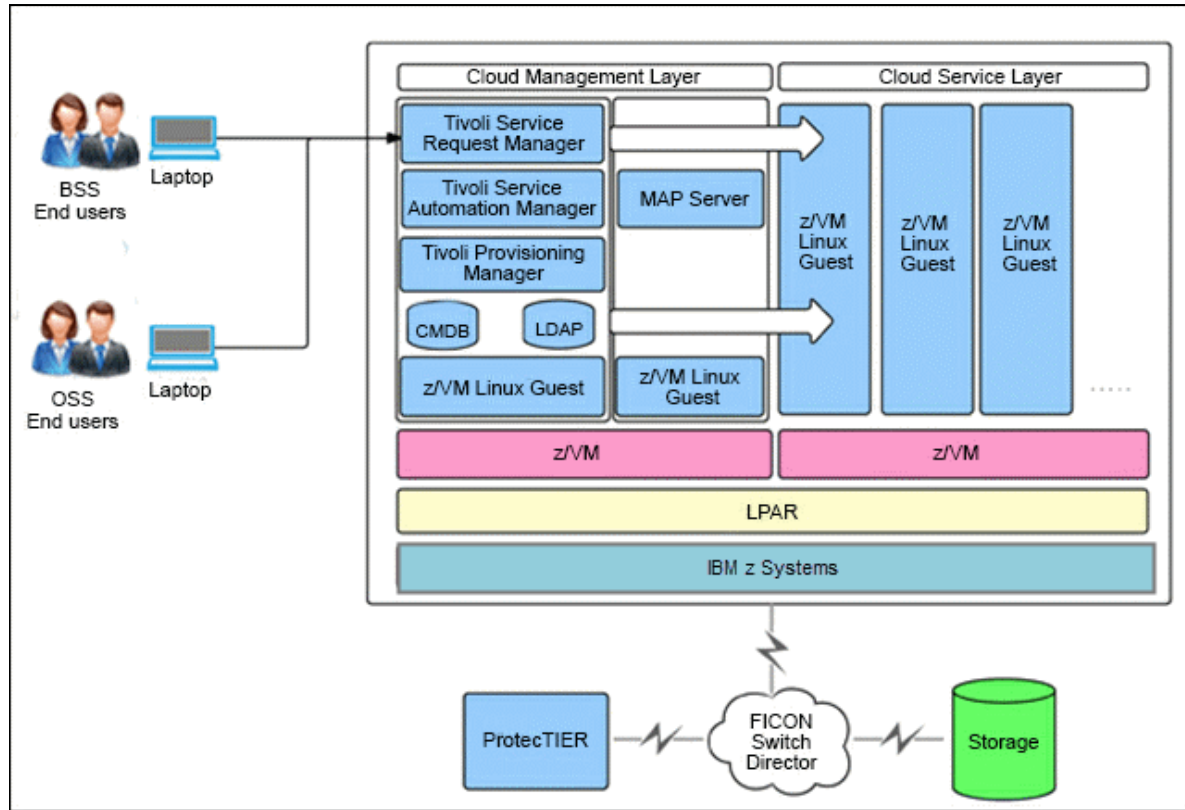


Figure A-1 Cloud solution that uses LinuxONE

Healthcare industry: Mobile and internet solution

In this scenario, Fictional Hospital H1 also chooses LinuxONE as its mobile application platform. Hospital H1 wants to build a secure platform, increase responsiveness and value perception, and reduce multi-platform development costs.

The following tool is used to 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.

The solution is designed to 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.

The solution also reduces 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-2 shows the secured access from a mobile device to a back-end transactional core system on the LinuxONE platform by using the global security policies and end-to-end secure transactions.

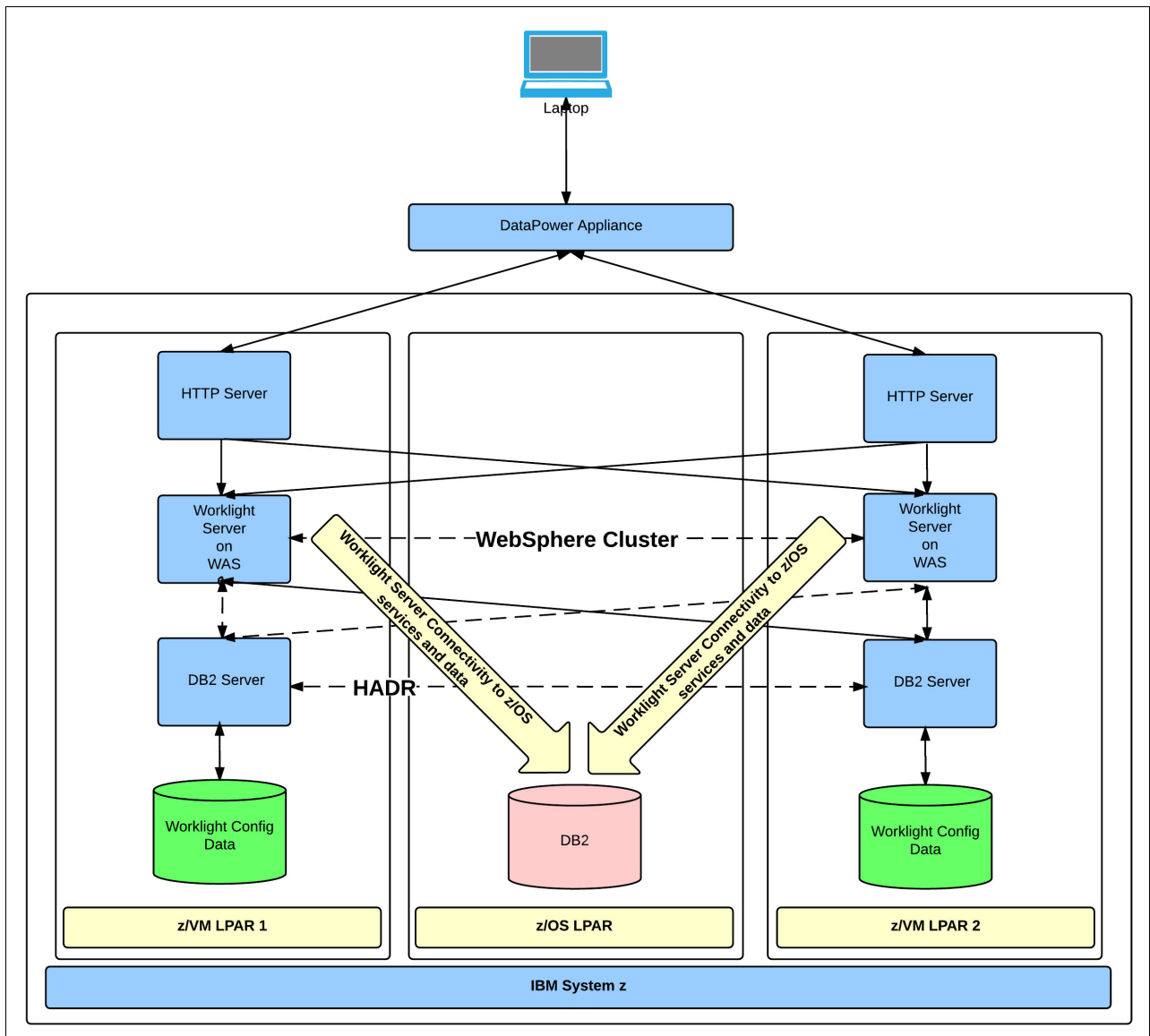


Figure A-2 Access from a mobile device to a back-end transactional core system

Energy and utilities industry: SAP Cloud and Automation solution on System z

In this scenario, Fictional Energy E1 chooses the System z platform as its SAP application running on Linux and database on IBM z/OS. Energy E1 wants to reduce the time spent to copy and refresh complete SAP systems from days to hours with a cloud solution and SAP system automation. This combination can automate, standardize, and increase the speed of day-to-day operations for SAP systems, reducing the risk of mistakes caused by human error.

The company also wants to reduce time spent on complex, repetitive tasks, freeing up skilled staff for higher value work and delivering higher operational efficiency. This reduction helps to slash costs and accelerate the time-to-value ratio for new workloads.

The company builds a virtual platform with the following items:

- ▶ IBM zEC12 or zBC12.
- ▶ IBM z/VM 6.3.
- ▶ Red Hat Enterprise Linux or SUSE Linux Enterprise Server on the System z platform.
- ▶ IBM DB2 for z/OS.
- ▶ IBM Database Provisioning System (DPS):
 - Web application JCL Engine.
 - Database Management.
 - Integrated with DB2 Cloning Tool.
- ▶ IBM DB2 Cloning Tool for z/OS: The DB2 Cloning Tool automates the cloning process to provide usable DB2 clones within minutes, boosting efficiency and freeing up DBA time:
 - Quickly clones DB2 subsystems, DB2 table spaces, or index spaces to create up-to-date test environments.
 - Automates the cloning process to provide usable DB2 clones within minutes.
 - Clones a DB2 subsystem by renaming and cataloging the data sets, fixing the volume internals, and updating the DB2 internal control information.
 - Fast copy technology quickly copies DB2 data sets within a subsystem or to a different subsystem.
 - Automates the cloning process using any volume level technology, such as IBM FlashCopy, to clone DB2 subsystems and any data set copy technology, such as FlashCopy, to clone table and index spaces and automatically convert the object IDs to simplify and automate the refresh of data.
- ▶ SAP NetWeaver Landscape Virtualization Management (LVM): By streamlining and automating critical business processes, SAP NetWeaver LVM software enables your IT department to focus on responding to new initiatives, controlling IT costs, and differentiating your business:
 - Manage your SAP landscape in physical and virtualized environments.
 - Central management point for your SAP landscape, start/stop, and mass operations.
 - Automate standard, day-to-day administrative and lifecycle management tasks.
 - Save time, effort, and money by automating copy, clone, and refresh.
- ▶ Build IBM Entry Cloud Solution for SAP with automated lifecycle management operations:

The IBM Entry Cloud Configuration solution automates complex tasks performed by administrators of databases, operating systems, storage systems, and SAP Basis. When combined with SAP NetWeaver LVM, the combination can reduce the time that it takes to copy and refresh complete SAP systems from days to hours. The high degree of automation also improves the quality and efficiency of SAP operations.

To build an IBM Entry Cloud solution for SAP, select automated lifecycle management operations such as:

 - SAP System Clone: Provision a fresh SAP system based on a new system copy.
 - SAP System Copy: Create a customized SAP system based on an existing system.

- SAP System Refresh: Copy DB content from PRD to Non-PRD including post processing.
- Create an additional dialog instance: Adding extra application server instances, for example, for monthly closing.

The Linux on System z, IBM Entry Cloud Configuration solution is the ideal productivity tool for any IT organization running SAP Business Suite on IBM zEnterprise® with IBM DB2 for z/OS. It is well-suited for computer services organizations that host SAP systems for their clients, and for any IT organization seeking to run its SAP operations with zEnterprise in an on-premises, self-managed, cloud computing environment. Figure A-3 shows the added value of this solution and how it reduces operation and administration time when compared to traditional operations.

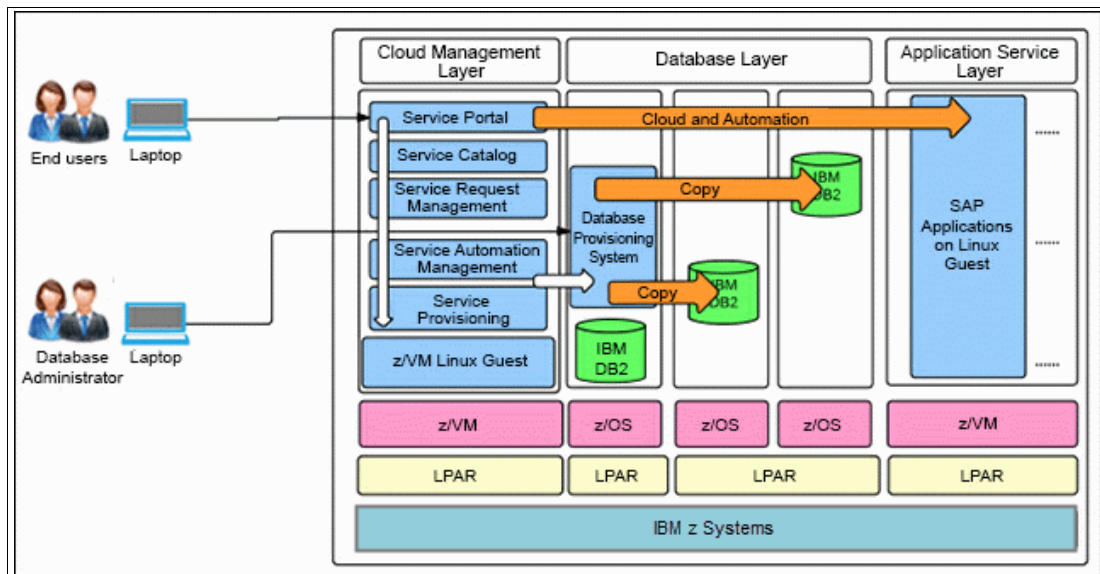


Figure A-3 Automate SAP System Copy with IBM Entry Cloud Solution for SAP

Related publications

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

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- ▶ *Advanced Networking Concepts Applied Using Linux on IBM System z*, SG24-7995
- ▶ *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 11gR2 on Linux on System z*, SG24-8104
- ▶ *Experiences with Oracle Solutions on Linux for IBM System z*, SG24-7634
- ▶ *IBM System Storage SAN Volume Controller and Storwize V7000 Best Practices and Performance Guidelines*, SG24-7521
- ▶ *IBM z Systems Connectivity Handbook*, SG24-5444
- ▶ *IBM Wave for z/VM Installation, Implementation, and Exploitation*, SG24-8192
- ▶ *IBM zEnterprise EC12 Technical Guide*, SG24-8049
- ▶ *Implementing FlashSystem 840 with SAN Volume Controller*, TIPS1137
- ▶ *Implementing the IBM System Storage SAN Volume Controller with IBM Spectrum Virtualize V7.6*, SG24-7933
- ▶ *Introduction to Storage Area Networks*, SG24-5470
- ▶ *Introduction to the New Mainframe: z/VM Basics*, SG24-7316
- ▶ *An Introduction to z/VM Single System Image (SSI) and Live Guest Relocation (LGR)*, SG24-8006
- ▶ *Linux on IBM eServer zSeries and S/390: Application Development*, SG24-6807
- ▶ *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
- ▶ *Using z/VM v 6.2 Single System Image (SSI) and Live Guest Relocation (LGR)*, SG24-8039
- ▶ *The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3*, SG24-8147

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:

- ▶ LinuxONE offers a variety of solutions
<http://www.ibm.com/systems/linuxone/solutions>
- ▶ GNU assembler manual
<http://www.gnu.org/software/binutils>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

- active task 110
- Active/Active 113
- archived data 93
 - incompatibility issues 94
- Availability
 - feature 108
 - scenarios 109

B

- backed-up need 91

C

- Capacity for a Planned Event (CPE) 109
- CBU
 - Capacity BackUp 109
- Collaborative Memory Management Assist (CMMA) 35
- concurrent user 84
- Confidentiality analysis 102
- Configuration file
 - syslog-ng 172
- Continuous Availability (CA) 107
- Continuous Operations (CO) 107
- Control 33
- Control Program (CP) 33
- Conversation Monitor System (CMS) 34
- Conversational Monitor System (CMS) 34
- Cooperative Memory Management (CMM) 35
- Customer Initiated Upgrade (CIU) 109

D

- data center 76, 107
- data migration 66
- database server 62, 75, 96
- DB2 data
 - replication feature 113
- DB2MOVE command 89
- DBA user 89
- designated IP address
 - Linux servers QDIO devices 66
- Disaster Recovery 107
 - predefined capacity 109
- Discontiguous Saved Segment (DCSS) 35
- disk device
 - access 89

E

- Enterprise authentication options 104
- Evaluation Acceptance Level (EAL) 96
- external firewall 64, 97

F

- Fibre Channel Protocol (FCP) 38
- file system 70, 75, 82, 86, 90
- firewall 96
- Firewalls and Existing Security Policies 96
- Firewalls and existing security policies 96
- Fixed Block Architecture (FBA) DASD 38
- FlashCopy 69

G

- Globally Dispersed Parallel Sysplex (GDPS) 115
- GNU Public License (GPL) 115
- golden image 70

H

- High Availability
 - Disaster Recovery (HADR) 113
- High Availability (HA) 107
- homemade applications 73

I

- IBM Tivoli System Automation 113
- incremental backup 92
- Infrastructure Service 79
- Integrated Cryptographic Service Facility (ICSF) 95
- Integrated Facility
 - for Linux 15
- Intellectual property 116
- IP address 66, 83–84, 101
- ISV 76
- ISV Application 47, 50

J

- Java Data Base Connector (JDBC) 114
- Java Virtual Machine (JVM) 81
- Just-In-Time (JIT) 81
- JVM switch 81

L

- Layer 2 57
- layer 2 VSWITCH 59
- Layer 3 57
- LDAP
 - user information 104
- Lightweight Directory Access Protocol (LDAP) 104
- Linux 49, 58, 66, 76, 84, 95
 - distribution 73, 93, 103
 - guest 59, 70, 85, 90
 - image 35, 96
 - kernel 37, 87
- Linux guest

- administration tasks 45
- log 33
- sizing process 91
- Linux kernel
- maintenance 108
- parameter 88
- Linux OS 70
- Linux Server 36, 90
 - insufficient memory size 90
- Linux system 34, 63, 69, 88
- Linux VM 99
 - internet 101
- Linux-HA Project 115
- Logical Partition 108
- Logical Volume
 - Manager 70
- LPAR 85, 108
- LVM device
 - file system size 73

M

- MAC address 59
- MediaWiki
 - software 154
- Memory Management features 35
- migration project 18, 44, 51, 106
 - execution stages 44
 - training considerations 45
 - various influences 45
- MS Windows 104
- Multiple LPARs 61, 111

N

- Named Saved Segment (NSS) 35

O

- On/Off Capacity on Demand 109
- Open Systems Adapter (OSA) 39, 96–97
- operating environment 43, 91, 94, 106
 - archived data 89
- operating system 48–49, 80, 94
 - target version 52
- Oracle database 78
- Oracle RAC 78, 114
- Oracle Real Application Clusters (RAC) 114
- OSA-Express2 card 58
- OSI Reference Model 57
- overcommitment ratio 35

P

- Payments Card Industry (PCI) 103
- physical firewall 63, 97
- principle of least privilege 95
- processor queue
 - number 86
- production stakeholders 104
- productive use
 - communications plan 48

- proof of concept (POC) 51, 80
- Public-Key Cryptography Standard
 - open source implementation 105
- Public-Key Cryptography Standard (PKCS) 105
- Publish your confidentiality policy 103

Q

- Queue Direct I/O (QDIO) 39

R

- real CPU 86
- reasonable cost 111
 - failure points 111
- Redbooks website 191
 - Contact us xi
- Rehosting Applications from Competitive Environments (RACE) 16
- response time 39, 44, 85, 116

S

- same LPAR 97, 114
 - virtual LAN 114
- Secure Sockets Layer
 - open source implementation 105
- Secure Sockets Layer (SSL) 102
- security through obscurity 96
- separate LPARs 60, 97
- server failure 107
- Service Level Agreement 51, 101, 116
- sharing same System z
 - different network segments 62
 - hardware 64
- single LPAR 85
 - multiple database servers 85
- Single point of control 69
- Single point of failure (SPOF) 108
- source application 20, 45, 70, 83, 105
- source environment 84
- source server 21, 66, 83–84
 - enough space 70
 - export/dump procedures 84
 - network file system 70
- source system 45
- staging server 94
- stakeholder 44
- stakeholders 44, 49, 97
- SUSE Linux
 - Enterprise Server 10.2 103
- SWAP device consideration 38
- System z 18, 21, 45–46, 106
 - application consolidation 18
 - environment 66
 - operating environment 18
 - power switch 106
 - server 68
 - swap device 38
 - virtual server 86

T

- tablespaces 89
- target environment 20, 50
 - assessing product 20
 - same characteristics 20
- target Linux 18, 66–67, 91
- target platform 16, 77
- target server 66–67, 83, 90
 - configure middleware 83
 - custom file system 83
 - file transfer process 70
 - performance test 83
- target system 70, 74, 90, 106, 116
- technology stakeholders 44

U

- UNIX administrator 45
- unneded process 35
- user acceptance testing
 - virtual servers 80

V

- V-DISK device 87
- virtual CPU 85
- virtual machine
 - complete System z environment 33
 - non-disruptive addition 110
 - z/VM directory 33
- virtual machine (VM) 33, 66, 110
- Virtual Machine Resource Manager (VMRM) 36
- VSWITCH 39, 57

W

- WebSphere application 110
- WebSphere Application Server setup 113
- wide area network (WAN) 46

Z

- z/VM layer 96
- z/VM maintenance 108
- z/VM session 33
 - Linux guest logs 33
- z/VM system 33
 - administrator 34
 - Virtual Network 108

Redbooks

Practical Migration from x86 to LinuxONE

(0.2"spine)
0.17" x 0.473"
90 x 249 pages



SG24-8377-00

ISBN 0738442283

Printed in U.S.A.

Get connected

