

Cloud Security Guidelines for IBM Power Systems

Turgut Aslan
Peter G. Croes
Liviú Rosca
Max Stern



 **Cloud**

Power Systems



International Technical Support Organization

Cloud Security Guidelines for IBM Power Systems

February 2016

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

Second Edition (February 2016)

This edition applies to IBM PowerVC 1.3.0 (5765-VCS), IBM PowerVM 2.2.4 (5765-PVS Standard Edition, 5765-PVE Enterprise Edition, 5765-PVL Linux Edition), IBM PowerKVM 3.1 (5765-KVM), IBM Cloud Manager with OpenStack 4.3 (5765-OSP), and the IBM Hardware Management Console 8.3.2 (7042-CR8).

© Copyright International Business Machines Corporation 2015, 2016. All rights reserved.

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

Contents

Notices	ix
Trademarks	x
IBM Redbooks promotions	xi
Preface	xiii
Authors	xiii
Now you can become a published author, too!	xv
Comments welcome	xv
Stay connected to IBM Redbooks	xvi
Part 1. Business context and architecture considerations	1
Chapter 1. Business context	3
1.1 Overview	4
1.1.1 Cloud deployment models	4
1.1.2 Cloud service models	5
1.2 Business drivers for cloud computing	6
1.3 IBM Power Systems and the cloud	7
1.3.1 Hypervisors	7
1.3.2 Platform management	8
1.3.3 Advanced virtualization management	8
1.3.4 Cloud management	9
1.4 Conclusion	11
Chapter 2. Cloud security reference architecture	13
2.1 IBM Cloud Computing Reference Architecture	14
2.1.1 Adoption patterns	15
2.1.2 Cloud Enabled Data Centers (or IaaS)	16
2.2 Security and the CCRA	18
2.2.1 Business drivers for a secure reference architecture	19
2.2.2 Security requirements	22
2.3 Cloud computing and regulatory compliance	24
2.3.1 Government regulations and agencies	24
2.3.2 Standards organizations	26
2.3.3 Industry bodies	27
2.3.4 Summary	28
2.4 Security guidance	28
2.4.1 Manage identities and access	29
2.4.2 Secure virtual machines	29
2.4.3 Patch default images	30
2.4.4 Manage logs and audit data	30
2.4.5 Network isolation	31
2.5 Usage scenarios	31
2.5.1 Generic use case with cloud-enabled data center	31
2.5.2 Typical PowerKVM use case	32
2.5.3 Typical PowerVM use case	33
2.6 Integration with IBM software	33
2.6.1 Security Information and Event Management (SIEM)	33

2.6.2 Identity and access management	34
2.6.3 Endpoint management	35
2.6.4 Threat and intrusion prevention	35
2.7 Conclusion	36
Part 2. Power cloud components	37
Chapter 3. IBM Hardware Management Console (HMC) security	39
3.1 Introduction to the HMC	40
3.2 User interfaces	40
3.3 Network interfaces	41
3.4 User and role management.	43
3.4.1 Users.	43
3.4.2 Roles.	44
3.4.3 Practical scenario of using users and customized roles	45
3.5 Monitoring and auditing HMC access	50
3.5.1 Access monitoring.	51
3.5.2 Access auditing.	51
3.6 Security enhancements and compliance	52
3.6.1 Security compliance	52
3.6.2 HMC security enhancements	52
3.6.3 Data replication	55
3.6.4 Customizing HMC encryption	55
3.7 HMC and security zones	56
3.7.1 Virtual switches	57
3.7.2 Enforcement of ACLs on virtual switches	59
3.7.3 ACL support for LPM	59
3.8 Conclusion	60
Chapter 4. IBM PowerVM security	61
4.1 IBM PowerVM overview	62
4.2 Isolation requirements for logical partitions.	62
4.2.1 Workload isolation.	62
4.2.2 Processor core isolation	63
4.2.3 Memory isolation.	63
4.2.4 I/O isolation	63
4.3 Domains of IBM Power processor cores.	63
4.3.1 Application domain	64
4.3.2 Kernel domain.	64
4.3.3 Hypervisor domain	64
4.4 Processor core access modes	65
4.5 POWER Hypervisor	65
4.5.1 POWER Hypervisor integrity.	66
4.5.2 POWER Hypervisor and processor core sharing	67
4.5.3 POWER Hypervisor and memory sharing.	68
4.5.4 POWER Hypervisor and I/O sharing.	68
4.6 Memory isolation	69
4.6.1 Effective memory	70
4.6.2 Virtual memory	70
4.6.3 Physical memory.	71
4.6.4 Real memory.	71
4.6.5 Logical memory.	72
4.6.6 Partition page tables	72
4.7 I/O isolation	73

4.8	Logical partitions (LPARs)	75
4.8.1	LPAR management	75
4.8.2	LPAR operating systems	76
4.9	Virtualization of I/O devices	76
4.9.1	Disk access for logical partitions	77
4.9.2	Network access for logical partitions	77
4.10	Security of DLPAR operations	79
4.11	IBM PowerVM security management with PowerSC	80
4.12	Secure Logical Partition Mobility	81
4.12.1	Live Partition Mobility	81
4.12.2	Practical scenario for secure LPM	82
4.13	PowerVM NovaLink	85
4.14	Conclusion	86
	Chapter 5. IBM PowerKVM security	87
5.1	PowerKVM architecture overview	88
5.1.1	PowerKVM host	89
5.1.2	PowerKVM guest	89
5.1.3	Quick Emulator (QEMU)	89
5.1.4	The libvirt library	89
5.1.5	The virsh virtualization shell tool	90
5.1.6	Kimchi	91
5.2	PowerKVM security considerations	92
5.2.1	Authentication	93
5.2.2	Networking	103
5.2.3	Firewall functionality with firewalld and iptables	116
5.2.4	Network filter driver	117
5.2.5	The sVirt service	120
5.2.6	Audit	125
5.2.7	PowerKVM guest image encryption	129
5.2.8	Guest live migration	134
5.3	Conclusion	136
	Chapter 6. IBM PowerVC security	137
6.1	Introduction to PowerVC and security topics	138
6.1.1	PowerVC architecture overview	139
6.1.2	Security enhancement features	143
6.1.3	Secure communications	151
6.2	Identity management	153
6.2.1	Removing the root account from the PowerVC admin group	154
6.2.2	PowerVC users, groups, roles, and policies	155
6.2.3	Using LDAP for PowerVC identity management	156
6.3	API security	158
6.3.1	Authentication	158
6.3.2	Secure communication for PowerVC APIs	161
6.3.3	Strict network access control	162
6.4	Audit	162
6.4.1	Enabling and disabling PowerVC audit	163
6.4.2	Retrieving audit log information	163
6.4.3	Important log files	165
6.5	Security options using <code>powervc-config</code> <i>command</i>	165
6.5.1	Setting the maximum image size	165
6.5.2	Setting the maximum amount of per-user image storage	166

6.6 Patch management	167
6.6.1 Where to get PowerVC security patch information	167
6.6.2 Consideration on OpenStack vulnerability	167
6.6.3 Managing Open Source components like Apache HTTP server, OpenSSL, and OpenSSH.	168
6.7 Conclusion	168
Chapter 7. IBM Cloud Manager with OpenStack security	169
7.1 Introducing IBM Cloud Manager with OpenStack	170
7.1.1 OpenStack and Chef.	170
7.1.2 Enhancements to OpenStack	170
7.1.3 Power Systems hypervisor support.	171
7.1.4 Deployment models	172
7.2 Identity.	173
7.2.1 Keystone and LDAP identities	173
7.2.2 Configuring LDAP	174
7.2.3 Projects, roles, and users	175
7.2.4 Changing default passwords.	176
7.2.5 Changing the default administrator user account	177
7.3 Access.	177
7.3.1 Access to provisioned virtual machines	178
7.3.2 Updating the default security policy	178
7.3.3 Generating and uploading SSH keys	181
7.3.4 Configuring SSL communication with self-service portal	183
7.3.5 Configuring SSL for OpenStack Dashboard	184
7.3.6 Network Time Protocol (NTP)	185
7.3.7 Session timeout and lockout	185
7.3.8 TCP/IP ports used by IBM Cloud Manager with OpenStack.	185
7.4 Patch management	187
7.5 Audit and logging	188
7.6 Image management	188
7.6.1 SSH host key entropy	189
7.6.2 Image staging project	189
7.7 REST API security	189
7.8 Conclusion	190
Chapter 8. IBM Bluemix secure gateway	191
8.1 IBM Bluemix overview.	192
8.1.1 How IBM Bluemix works	193
8.1.2 IBM Bluemix management	194
8.2 IBM Bluemix Secure Gateway	195
8.2.1 IBM Bluemix Secure Gateway configuration.	196
8.2.2 IBM Bluemix Secure Gateway service status	201
8.3 Other security options of IBM Bluemix	202
Part 3. Appendixes	205
Appendix A. Troubleshooting SSL and TLS handshake	207
Collecting network data by using tcpdump.	208
Examining packet captures with Wireshark	208
Introduction to SSL and TLS handshake	208
Examining the SSL or TLS handshake	209
Other tools	213

Appendix B. VMware vRealize Automation for Power Systems 215

Related publications 219

IBM Redbooks 219

Online resources 219

Help from IBM 222

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

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

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

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

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

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

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

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

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


COPYRIGHT LICENSE:

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

Trademarks

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

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

Active Memory™	IBM SmartCloud®	PowerVM®
AIX®	IBM Spectrum™	QRadar®
AppScan®	IBM z Systems™	Redbooks®
BigFix®	Micro-Partitioning®	Redbooks (logo)  ®
Bluemix®	POWER®	SiteProtector™
DataPower®	POWER Hypervisor™	Storwize®
DB2®	Power Systems™	System Storage®
DS8000®	POWER7+™	Tivoli®
Global Technology Services®	POWER8®	X-Force®
IBM®	PowerHA®	XIV®
IBM Cloud Managed Services™	PowerSC™	z/VM®

The following terms are trademarks of other companies:

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

ITIL is a Registered Trade Mark of AXELOS Limited.

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.

Find and read thousands of IBM Redbooks publications

- ▶ Search, bookmark, save and organize favorites
- ▶ Get personalized notifications of new content
- ▶ Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App



Download
Now

Android



Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!



ibm.com/Redbooks

About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redbooks® publication is a comprehensive guide that covers cloud security considerations for IBM Power Systems™. The first objectives of this book are to examine how Power Systems can fit into the current and developing cloud computing landscape and to outline the proven Cloud Computing Reference Architecture (CCRA) that IBM employs in building private and hybrid cloud environments.

The book then looks more closely at the underlying technology and hones in on the security aspects for the following subsystems:

- ▶ IBM Hardware Management Console
- ▶ IBM PowerVM®
- ▶ IBM PowerKVM
- ▶ IBM PowerVC
- ▶ IBM Cloud Manager with OpenStack
- ▶ IBM Bluemix®

This publication is for professionals who are involved in security design with regard to planning and deploying cloud infrastructures using IBM Power Systems.

Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Rochester, MN Center.



Turgut Aslan is an Architect and Development Workstream Leader for security and compliance in IBM Cloud Managed Services™ (CMS), working at IBM Research and Development in Boeblingen, Germany. Turgut holds a PhD degree in physics from the Erlangen-Nuernberg University, Germany. He has 15 years of experience in security processes, tools, solutions, risk management, regulatory compliance, and audit. Previously Turgut participated in two ITSO social media residencies and extensively writes blogs about cyber security, data privacy, big data, and cloud.



Peter G. Croes is an Open Group Certified Advisory Client Technical Specialist working for IBM Systems Hardware with the European team providing pre-sales support. He has over 25 years of experience in a variety of areas that are related to infrastructure and solution design, virtualization, and cloud computing. His expertise is related to solutions leveraging IBM Power Systems, virtualization, IBM i, and Linux. He wrote an IBM i Virtualization white paper that is recognized as the leading document to use for both IBM Business Partners and clients and is referenced worldwide during events. Peter is a regular speaker at IBM Technical Universities and Common events and is also an author of several other Redbooks publications.



Liviu Rosca is a Senior Product Support and IT Specialist with IBM Global Technology Services® in Romania. He has 14 years of experience in Power Systems, IBM AIX®, and IBM PowerHA®. His areas of expertise include Power Systems, AIX, PowerHA, networking, security, and telecommunications. He teaches AIX and PowerHA classes. He has co-authored other Redbooks publications.



Max Stern is a Technical Consultant at IBM Moscow, working in the Lab Services IBM EE/A for 5 years. He focuses on UNIX performance, security, and cloud technologies. Max is a member of the global IBM Power Care on performance and security and has skills in AIX, Linux, Solaris, high availability solutions, enterprise hardware, SAN, and storage. He is a Certified IBM Advanced Technical Expert and Certified Sun Solaris Certified Advanced System Administration. Prior to working at IBM, Max worked at Nokia Siemens Networks as a UNIX System Engineer and Database Administrator.

This second edition of this IBM Redbooks project was led by:

Debbie Landon

International Technical Support Organization, Rochester Center

Thanks to the authors of the first edition of this book:

- ▶ Axel Buecker, IBM USA
- ▶ Thiago Costa, IBM Brazil
- ▶ Mu Hyun Kim, IBM Korea
- ▶ Liviu Rosca, IBM Romania
- ▶ Stephen Tremain, IBM Australia

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

Bruce Anthony, Shamsunder Ashok, Thomas Bosworth, Rosa Davidson, Saurabh Desai, Jason Furmanek, Christopher Hales, Pete Heyrman, John Jacobson, George Koikara, Heather Kreger, Jay Kruemcke, Vess Natchev, Naresh Nayar, Dino Quintero, Jeffrey Schaefer, Anna Sortland, Alise Spence, Michael Strosaker, Debora Velarde, Allyn Walsh, George Wilson, Kyle Wurgler

IBM

David Bennin, Rich Conway, Ann Lund

International Technical Support Organization, Poughkeepsie Center

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
3605 Hwy 52 North
Rochester, MN 55901

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

Business context and architecture considerations

This part describes how IBM Power Systems can fit into the current and developing cloud computing landscape. It also outlines the Cloud Computing Reference Architecture (CCRA) that IBM employs in building cloud solutions for clients. Of particular focus are the security aspects of the CCRA and how they relate to Power Systems in the cloud.

This part contains the following chapters:

- ▶ Chapter 1, “Business context” on page 3
- ▶ Chapter 2, “Cloud security reference architecture” on page 13



Business context

IBM Power Systems are uniquely suited for cloud environments, with their industry-leading virtualization, enterprise-class security, elastic scalability and reliability, availability, and serviceability.

This chapter describes how Power Systems fit into the current and developing cloud computing landscape.

The following topics are covered in this chapter:

- ▶ 1.1, “Overview” on page 4
- ▶ 1.2, “Business drivers for cloud computing” on page 6
- ▶ 1.3, “IBM Power Systems and the cloud” on page 7
- ▶ 1.4, “Conclusion” on page 11

1.1 Overview

As with most new technology paradigms, security concerns surrounding cloud computing have become the most widely discussed inhibitor of widespread usage, whether for public, private, or hybrid cloud installations.

To gain the trust of organizations, cloud services must deliver security and privacy expectations that meet or exceed what is available in traditional IT environments. In fact, because of the ease in which virtual machines can be deployed, security management should aim to exceed that of the traditional IT environment.

1.1.1 Cloud deployment models

The National Institute of Standards and Technology (NIST) provides the following definition for cloud computing¹:

“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

Cloud computing can be provided in several delivery or deployment models:

- Public clouds

A public cloud is one in which the cloud infrastructure is made available to the general public or a large industry group over the Internet. The infrastructure is not owned by the user, but by an organization providing cloud services, hosting it on premises. Services can be provided either at no cost, as a subscription, or under a pay-as-you-go model.

- Private clouds

A private cloud refers to a cloud solution where the infrastructure is provisioned for the exclusive use of a single organization. The organization often acts as a cloud service provider to internal business units that obtain all the benefits of a cloud without having to provision their own infrastructure. By consolidating and centralizing services into a cloud, the organization benefits from centralized service management and economies of scale.

A private cloud can provide an organization with some advantages over a public cloud. The organization gains greater control over the various resources that make up the cloud. In addition, private clouds are ideal when the type of work being done is not practical for a public cloud because of network latency, security, or regulatory concerns.

The organization, a third party, or a combination can own, manage, and operate a private cloud. The private cloud infrastructure is usually provisioned on the organization's premises, but it can also be hosted in a data center that is owned by a third party.

- Hybrid clouds

A hybrid cloud, as the name implies, is a combination of various cloud types (public, private, or community). Each cloud in the hybrid mix remains a unique entity, but is bound to the mix by technology that enables data and application portability.

The hybrid approach allows a business to take advantage of the scalability and cost-effectiveness of a public cloud without exposing applications and data beyond the corporate intranet.

¹ NIST Special Publication 800-145, P. Mell, T. Grance. *The NIST Definition of Cloud Computing*:
<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>

A well constructed hybrid cloud can service secure, mission-critical processes, such as receiving customer payments (a private cloud service), and also those that are secondary to the business, such as employee payroll processing (a public cloud service).

- **Community clouds**

A community cloud shares the cloud infrastructure across several organizations in support of a specific community that has common concerns (for example, mission, security requirements, policy, and compliance considerations). The primary goal of a community cloud is to have participating organizations realize the benefits of a public cloud, such as shared infrastructure costs and a pay-as-you-go billing structure, with the added level of privacy, security, and policy compliance usually associated with a private cloud.

The community cloud infrastructure can be provided on premises or at a third-party data center, and can be managed by the participating organization or a third party.

1.1.2 Cloud service models

In addition, the industry recognizes three cloud computing service models (IaaS, PaaS, SaaS), although others exist. These are briefly described here, but for further details, see the NIST publication¹ and explore the web.

- **Infrastructure as a service (IaaS)**

The delivery of a computer infrastructure, including server functionality, networking functionality, data center functionality, and storage functionality as an outsourced service.

- **Platform as a service (PaaS)**

PaaS is the delivery of a computing platform, including applications, optimized middleware, development tools, and Java and Web 2.0 runtime environments, in a cloud-based environment.

- **Software as a service (SaaS)**

SaaS is a model of software deployment whereby software including business processes, enterprise applications, and collaboration tools, are provided as a service to customers through the cloud.

Figure 1-1 on page 6 shows the service responsibility line (SRL), which defines the upper service management level of the cloud service provider (CSP) for IaaS, PaaS, and SaaS

	IaaS	PaaS	SaaS
SRL	Business Process	Business Process	Business Process
	Applications	Applications	Applications
	Data	Data	Data
	Runtime	Runtime	Runtime
	Middleware	Middleware	Middleware
	O/S	O/S	O/S
	Virtualization	Virtualization	Virtualization
	Servers	Servers	Servers
	Storage	Storage	Storage
	Networking	Networking	Networking

Figure 1-1 Cloud service provider Service Responsibility Line (SRL) for IaaS, PaaS, and SaaS

1.2 Business drivers for cloud computing

Interest in cloud computing is currently strong among today's Chief Information Officers (CIO) and business leaders. An IBM study² found that the cloud's strategic importance to decision-makers, such as Chief Executive Officers (CEO), Chief Marketing Officers (CMO), finance, human relations and procurement executives, is poised to double from 34% to 72%. The survey found that one out of five organizations is ahead of the curve on cloud adoption and achieving competitive advantage, not just cutting costs and driving efficiency, through cloud computing.

Pacesetters in cloud adoption have reported impressive competitive advantages from cloud computing, but the journey has not been easy. Many respondents (44%) believe cloud introduces greater complexity into their organization.

To manage this growing complexity, early adopters are more likely to employ these approaches:

- ▶ Have an enterprise-wide cloud strategy
- ▶ Favor open source cloud platforms
- ▶ Use a hybrid cloud
- ▶ Have executive support for cloud experimentation

² Under cloud cover: How leaders are accelerating competitive differentiation, IBM Center for Applied Insights, 2013: <http://www.ibm.com/press/us/en/pressrelease/42304.wss> (At this site, click the **IBM survey** link.)

Key drivers for business adoption of cloud computing are as follows:

- ▶ Lower the costs and improve the economics of delivering IT solutions.
- ▶ Use cloud infrastructure with confidence that it is secure, compliant, and meets regulatory requirements.
- ▶ Leverage existing investment and extend current infrastructure to implement security for virtual infrastructure.
- ▶ Decrease delivery and provisioning time for secure new services through standardization and automation.
- ▶ Maintain service level compliance, accuracy, repeatability and traceability for the cloud environment.
- ▶ Align IT resource allocation with business goals.

1.3 IBM Power Systems and the cloud

A cloud computing environment built with IBM Power Systems can help organizations transform their data centers to become cloud-ready.

IBM Power Systems are uniquely suited for cloud environments with their industry-leading virtualization, enterprise-class security, elastic scalability and reliability, availability, and serviceability (RAS). Power Systems provide the necessary memory bandwidth and compute power to deliver performance that big data, analytics, and other compute-intensive cloud services require.

The following topics are covered in this section:

- ▶ 1.3.1, “Hypervisors” on page 7
- ▶ 1.3.2, “Platform management” on page 8
- ▶ 1.3.3, “Advanced virtualization management” on page 8
- ▶ 1.3.4, “Cloud management” on page 9

1.3.1 Hypervisors

Power Systems provide flexibility with choices of operating system, hypervisor, and cloud management solutions using OpenStack and based on open standards.

IBM Power Systems provide a choice of hypervisor with IBM PowerVM or the open source IBM PowerKVM.

PowerVM

PowerVM makes efficient use of system resources, which are built directly into the firmware of all Power Systems. Being part of the firmware also means PowerVM is secure by design, with zero reported common vulnerabilities exposures for the PowerVM hypervisor³ in the US Government NIST database⁴.

PowerVM supports AIX, IBM i, and Linux on Power as guest operating systems.

For more information about PowerVM, see Chapter 4, “IBM PowerVM security” on page 61.

³ *Is your platform secure? Really?:* <http://www.ibm.com/systems/power/migratetoibm/assets/security.html>

⁴ NIST database: <http://nvd.nist.gov/>

PowerKVM

PowerKVM presents a choice for customers looking for a low cost, open source virtualization solution for Linux on Power Systems built on the IBM POWER8® architecture. PowerKVM provides support for Red Hat Enterprise Linux, SUSE Linux Enterprise Server, and Ubuntu guests, and is optimized for the Power Systems platform with support for processor and memory-sharing-over-commit for higher utilizations, dynamic addition and removal of virtual devices, and live virtual machine (VM) migration.

PowerKVM is truly open source. The code and its repository data are available, continuously inspected, and transparent in modification rationale throughout the product lifecycle.

For more information about PowerVM, see Chapter 5, “IBM PowerKVM security” on page 87.

1.3.2 Platform management

Platform management is usually implemented using a web interface, and allows basic interaction with the hypervisor, allowing for such tasks as creation and deletion of VMs (logical partitions, LPARs) and management of VMs.

IBM Hardware Management Console

The IBM Hardware Management Console (HMC) is a Linux based appliance for planning, deploying and managing IBM Power Systems.

With the HMC, a system administrator can do logical partition functions, service functions, and various system management functions by using either the web browser interface, or the command-line interface (CLI).

See the following sources for more information:

- ▶ The functionality of the HMC extends beyond the basic administration of LPARs, and is described in *IBM Power Systems HMC Implementation and Usage Guide*, SG24-7491.
- ▶ The latest enhancements of HMC, including updates to NIST support, are described in *IBM Power Systems Hardware Management Console: Version 8 Release 8.1.0 Enhancements*, SG24-8232.
- ▶ More information about the HMC is also in Chapter 3, “IBM Hardware Management Console (HMC) security” on page 39.

Kimchi

Kimchi is an open-source management layer for PowerKVM built upon open standards such as libvirt. Kimchi provides a simple to use web interface for managing PowerKVM, and virtual machines, built upon a RESTful API.

For more information about Kimchi, see 5.1.6, “Kimchi” on page 91.

1.3.3 Advanced virtualization management

Advanced virtualization tools build on the basic capabilities of platform management tools, and enhance the management of resources, images, and deployments.

Advanced virtualization management is provided by PowerVC (Virtualization Center). Built on OpenStack, it allows Power System cloud infrastructure to plug into a broad array of management solutions.

With PowerVC, the system administrator can perform the following activities:

- ▶ Create virtual machines and resize their CPU and memory.
- ▶ Attach disk volumes to the virtual machines.
- ▶ Import existing virtual machines and volumes so they can be managed by PowerVC.
- ▶ Monitor the utilization of the resources that are in your environment.
- ▶ Migrate virtual machines while they are running (live migration between physical servers).
- ▶ Deploy images quickly to create new virtual machines that meet the demands of dynamic business needs. PowerVC can deploy virtual machines running IBM AIX, IBM i, or Linux operating systems.

PowerVC provides advanced virtualization management by leveraging the HMC or the PowerVM NovaLink REST API when managing PowerVM. However, it communicates directly with PowerKVM because PowerKVM does not support the HMC. IBM Cloud Manager with OpenStack offerings require PowerVC to manage PowerVM configurations.

For more information about PowerVC, see Chapter 6., “IBM PowerVC security” on page 137.

1.3.4 Cloud management

Cloud management layers are differentiated from advanced virtualization management layers by the ability to manage hypervisors on different platforms, or even from different vendors.

Functionality is also extended to include self-service interfaces, resource monitoring, image catalog, and more advanced features.

The IBM cloud management solutions are based on OpenStack technology. Key benefits of OpenStack are as follows:

- ▶ Large development community with continued rapid growth.
- ▶ Open and extensible architecture to quickly integrate into existing infrastructures.
- ▶ Broad industry support and ecosystem for extensive device support and cloud standards.
- ▶ Open APIs provide flexibility and agility. Access to open APIs can help reduce vendor lock-in to a particular hypervisor or platform.

OpenStack: OpenStack is a global collaboration of developers and cloud computing technologists working to produce a ubiquitous infrastructure as a service (IaaS) open source cloud computing platform for public and private clouds. IBM is a platinum sponsor of the OpenStack Foundation. More information is available at the OpenStack website:

<http://www.openstack.org/>

IBM Cloud Manager with OpenStack

IBM Cloud Manager with OpenStack is a modular, highly flexible, and easy-to-use solution designed to deliver cloud services for private or public clouds. Cloud administration is simplified through an intuitive interface for managing projects, users, and applications and also monitoring heterogeneous workloads and cloud resources.

Several key features of Cloud Manager with OpenStack are as follows:

- ▶ Simplified user self-service portal for automated delivery of requested services without IT intervention.
- ▶ Virtualized image management with library for standardized images, automated approval processing, and provisioning.
- ▶ Full OpenStack API support for interoperability and customization to help tailor services to unique business requirements.
- ▶ Basic metering to support allocation of IT infrastructure usage to appropriate individuals, groups, or departments.
- ▶ Heterogeneous support for IBM PowerVM, IBM z/VM®, IBM PowerKVM, x86 KVM, Microsoft Hyper-V and VMware vSphere virtualization environments.
- ▶ Support for live migrations from one host to another within a OpenStack cloud region.

Cloud Manager with OpenStack also provides an on-ramp to more advanced IBM cloud offerings, such as IBM Cloud Orchestrator.

For more information about Cloud Manager with OpenStack, see Chapter 7, “IBM Cloud Manager with OpenStack security” on page 169.

IBM Cloud Orchestrator

IBM offers solutions to deliver advanced cloud capabilities such as rapid and scalable provisioning, robust workflow orchestration, virtualization lifecycle management, and sophisticated billing and chargeback with IBM Cloud Orchestrator.

These solutions, built on open source solutions including OpenStack and Chef, are designed to reduce the risk associated with software integration and accelerate delivery of advanced cloud computing capabilities.

Key features of IBM Cloud Orchestrator are as follows:

- ▶ Pattern-based cloud delivery
- ▶ Business process designing
- ▶ Monitoring dashboard
- ▶ Enhanced multitenancy
- ▶ High availability
- ▶ Separate self-service user interface and administrative user interface
- ▶ Cost management
- ▶ Full access to OpenStack Icehouse command-line interface, including Nova, Glance, Keystone, Cinder, Neutron and Horizon modules
- ▶ Heterogeneous support for IBM PowerVM, IBM z/VM, IBM PowerKVM, x86 KVM, Microsoft Hyper-V and VMware vSphere virtualization environments

An overview of the various hypervisors, platform management, advanced virtualization management, and cloud management tools are depicted in Figure 1-2.

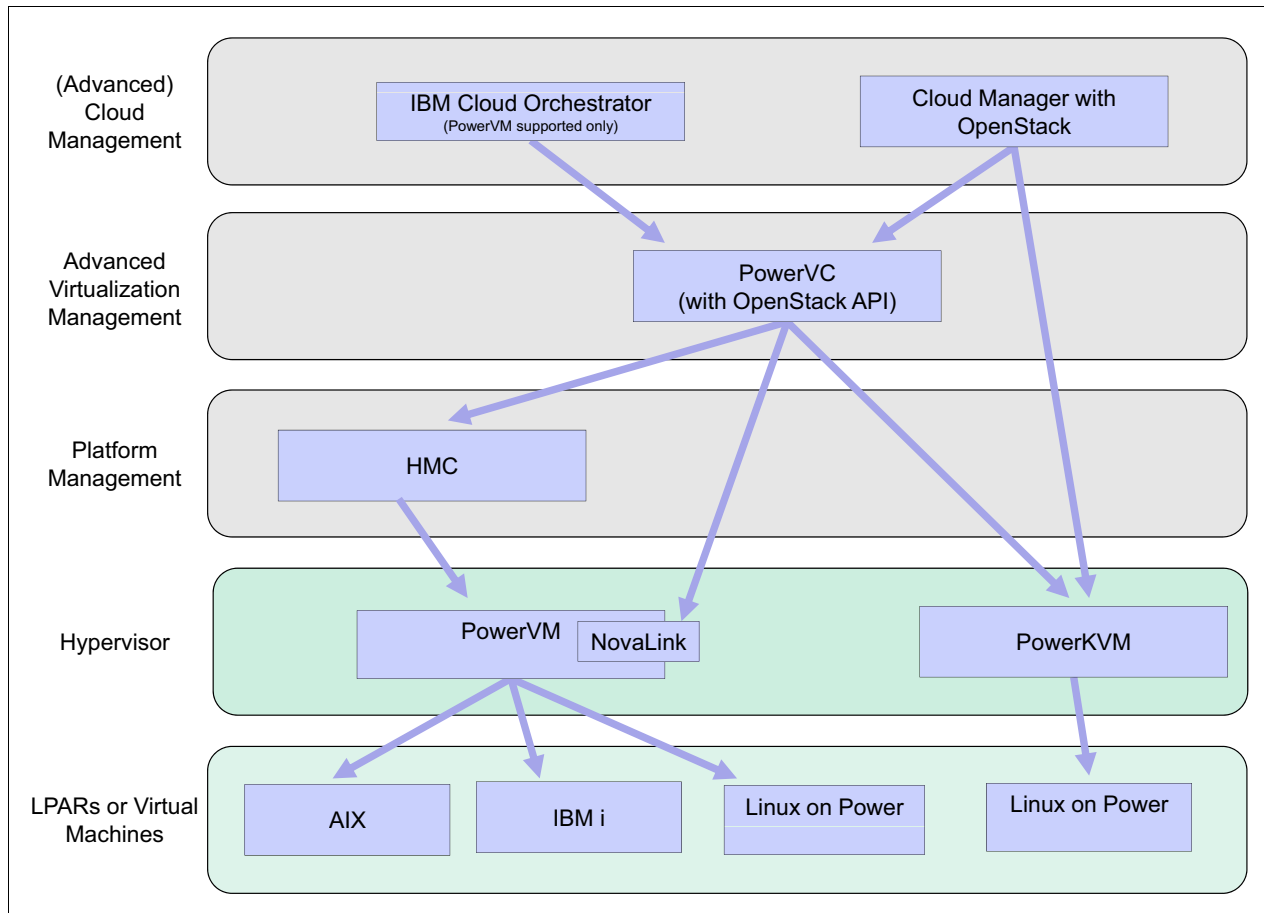


Figure 1-2 Overview of IBM Power Systems management options

1.4 Conclusion

Power Systems cloud solutions can help organizations reduce IT costs, improve service delivery, and encourage business innovation, with an effective and efficient cloud computing environment, which is as follows:

- ▶ Dynamic for automated, optimum resource allocation and superior economics
- ▶ Scalable for your smallest and largest workloads
- ▶ Reliable for high qualities of service across the cloud
- ▶ Flexible with open-source-based tools and APIs for maximum customizability



Cloud security reference architecture

Cloud computing can potentially be a disruptive change to the way an organization's IT services are delivered. Without sound planning and design, the complexity of cloud computing can be inhibitory to business adoption.

This chapter outlines the proven Cloud Computing Reference Architecture (CCRA) that IBM both employs in building private clouds for clients and also uses in the IBM owned public and private clouds. This chapter particularly focuses on the security aspects of the CCRA, and how they relate to Power Systems in the cloud.

The following topics are covered in this chapter:

- ▶ 2.1, "IBM Cloud Computing Reference Architecture" on page 14
- ▶ 2.2, "Security and the CCRA" on page 18
- ▶ 2.3, "Cloud computing and regulatory compliance" on page 24
- ▶ 2.4, "Security guidance" on page 28
- ▶ 2.5, "Usage scenarios" on page 31
- ▶ 2.6, "Integration with IBM software" on page 33
- ▶ 2.7, "Conclusion" on page 36

2.1 IBM Cloud Computing Reference Architecture

The IBM Cloud Computing Reference Architecture (CCRA) provides prescriptive guidance for preferred approaches to building and deploying cloud computing implementations.

The CCRA represents the aggregate experience from hundreds of cloud client engagements and IBM hosted cloud implementations. It is based on knowledge of IBM services, software, and system experiences, including IBM Research. The continuous improvement model ensures that both real-world experience and technology advancements from IBM Research are integrated into the CCRA.

An important aspect is that the CCRA approach to security is to be secure by design, and focus on building security into the fabric of the cloud.

Figure 2-1 shows a high-level diagram of the CCRA with the three major roles of the cloud service consumer, cloud service provider, and cloud service creator.

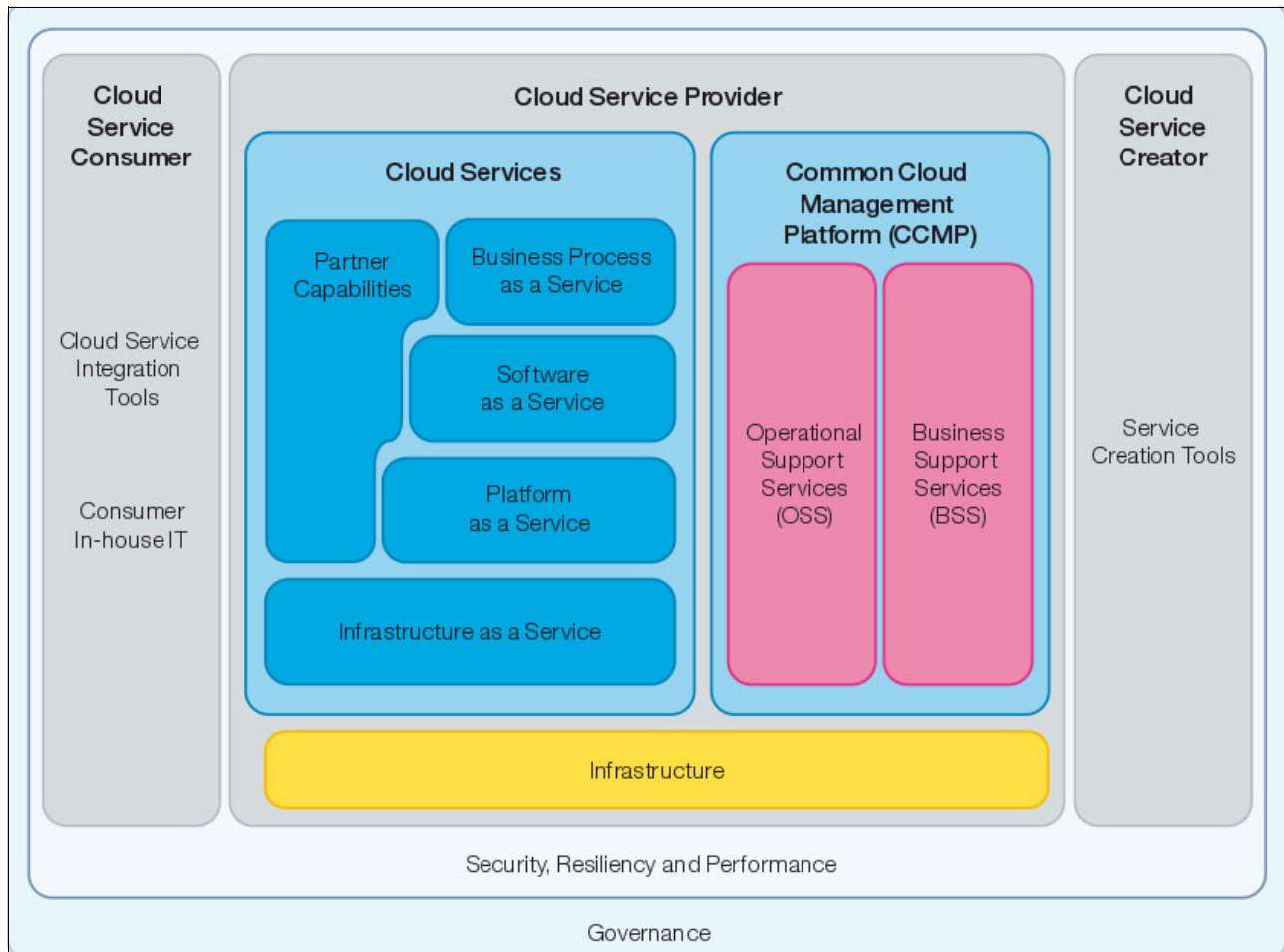


Figure 2-1 IBM CCRA with roles and cloud service provider details

Cloud computing must be enabled with effective security, resiliency, service management, governance, business planning, and lifecycle management. By delivering best practices in a standardized methodical way, this reference architecture can ensure consistency and quality across IBM development and delivery projects.

The IBM Cloud Computing Reference Architecture offers these benefits:

- ▶ Is built on open standards.
- ▶ Delivers robust security, governance, and privacy capabilities.
- ▶ Combines powerful automation and services management with rich business management functions for fully integrated, top-to-bottom management of cloud infrastructure and cloud services.
- ▶ Supports the full spectrum of cloud service models, including software as a service (SaaS), platform as a service (PaaS), infrastructure as a service (IaaS), and cloud service providers (CSP).
- ▶ Enables the flexible scaling and resiliency required for successful cloud economics and return on investment.
- ▶ Facilitates seamless integration into existing environments.
- ▶ Is based on industry-leading expertise with service-oriented architecture (SOA) for building services and SOAs.

For more information, see “Getting cloud computing right,” which is an IBM Global Technology Services Thought Leadership White Paper:

<http://www.ibm.com/de/cloud/pdf/Gettingcloudcomputingright.pdf>

2.1.1 Adoption patterns

The CCRA identifies four cloud adoption patterns:

- ▶ Cloud Enabled Data Center (IaaS)
The Cloud Enabled Data Center adoption pattern is typically the entry point into the cloud solutions space. It provides guidance on the definition, design, and deployment of cloud-computing solutions that deliver IaaS typically within the enterprise boundaries.
- ▶ Platform services (PaaS)
The PaaS adoption pattern describes how to design cloud-computing solutions that deliver preconfigured ready-to-execute runtime environments or middleware stacks onto which applications can be deployed. It also describes how to tie together application development and application deployment processes into a single continuous delivery process based on application development and IT operations (DevOps) principles.
- ▶ Software services (SaaS)
The SaaS adoption pattern defines the architecture for definition and operation of SaaS applications. The Cloud Service Provider adoption pattern provides the architecture that enables SaaS applications to be managed and offered by the cloud service provider. The cloud service provider also supports systems that provide the environment in which SaaS business models are realized.
- ▶ Cloud Service Provider
The Cloud Service Provider adoption pattern defines cloud-based solutions that provide cloud services through a service provider model. A service provider is an organization that provides the cloud usually for external customers. A service provider manages and provides cloud services as a general provider, rather than operating a computing facility for its own organization.

The CCRA supports the implementation of cloud services as a private cloud (site on-premises), public cloud (site on CSP), or a combination of the two (hybrid cloud), shown in Figure 2-2.

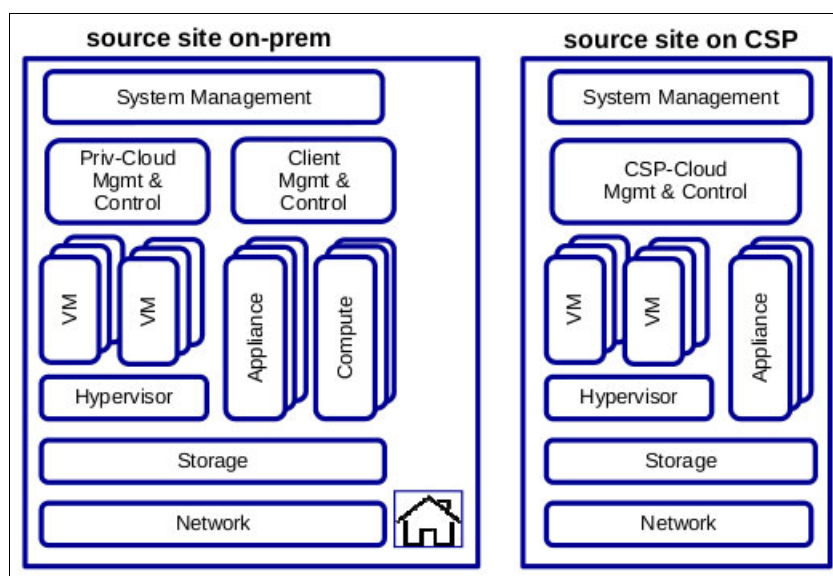


Figure 2-2 Private cloud (on-premises site) and public cloud (on cloud service provider site)

Infrastructure as a service (IaaS) is the typical starting point for most organizations when moving to a cloud computing environment. IaaS can be used for the delivery of resources such as compute, storage, and network services through a self-service portal. As such, this book focuses on suggestions for secure implementations of IBM Power Systems in a *private cloud enabled data center* or *IaaS cloud*.

Note: When implementing a private cloud, you have several choices. You can build it yourself on-premises, or you can contract a cloud service provider (CSP) to provide it for you. If you are evaluating CSPs, the Cloud Security Alliance (CSA) has several documents that allows you to assess the potential security risks associated with CSPs at this link:

<https://cloudsecurityalliance.org>

2.1.2 Cloud Enabled Data Centers (or IaaS)

The adoption pattern for IaaS as defined by the CCRA is called the *Cloud Enabled Data Center* adoption pattern. The Cloud Enabled Data Center adoption pattern contains prescriptive guidance on how to architect, design, and implement an IaaS solution.

Because IaaS is often the starting point, IaaS must be supported by a modular and flexible architecture that easily allows the integration of more capabilities and more robust capabilities. The Cloud Enabled Data Center adoption pattern provides the necessary modular architecture to accomplish this goal by establishing the architectural framework for designing IaaS solutions.

The CCRA identifies an incremental approach for building Cloud Enabled Data Centers. As shown in the following list, the cloud infrastructure can start by providing basic virtualization and image management capabilities, but can be enhanced with more sophisticated features, such as event management and capacity planning, over time.

- ▶ Simple IaaS services
 - VM provisioning and on-boarding
 - Cloud management
 - Role and authentication management
 - VM image construction
 - Image management
 - Usage metering, accounting and chargeback
- ▶ Cloud management
 - Virtualized infrastructure monitoring
 - Capacity management and planning
 - Event management
 - Backup and restore
 - Patches management
 - Endpoint compliance and management
- ▶ Advanced IaaS services
 - Storage provisioning and automation management
 - Network provisioning and automation management
 - Services orchestration
 - Hybrid cloud integration
 - Threat and vulnerability management
 - Identity and access management
 - Security info and events management
- ▶ ITIL managed IaaS services
 - Problem and incident management
 - IT asset management
 - License management
 - Change and configuration management
 - Service desk
 - Release management

The increasing capabilities of the Cloud Enabled Data Center are shown in Figure 2-3.

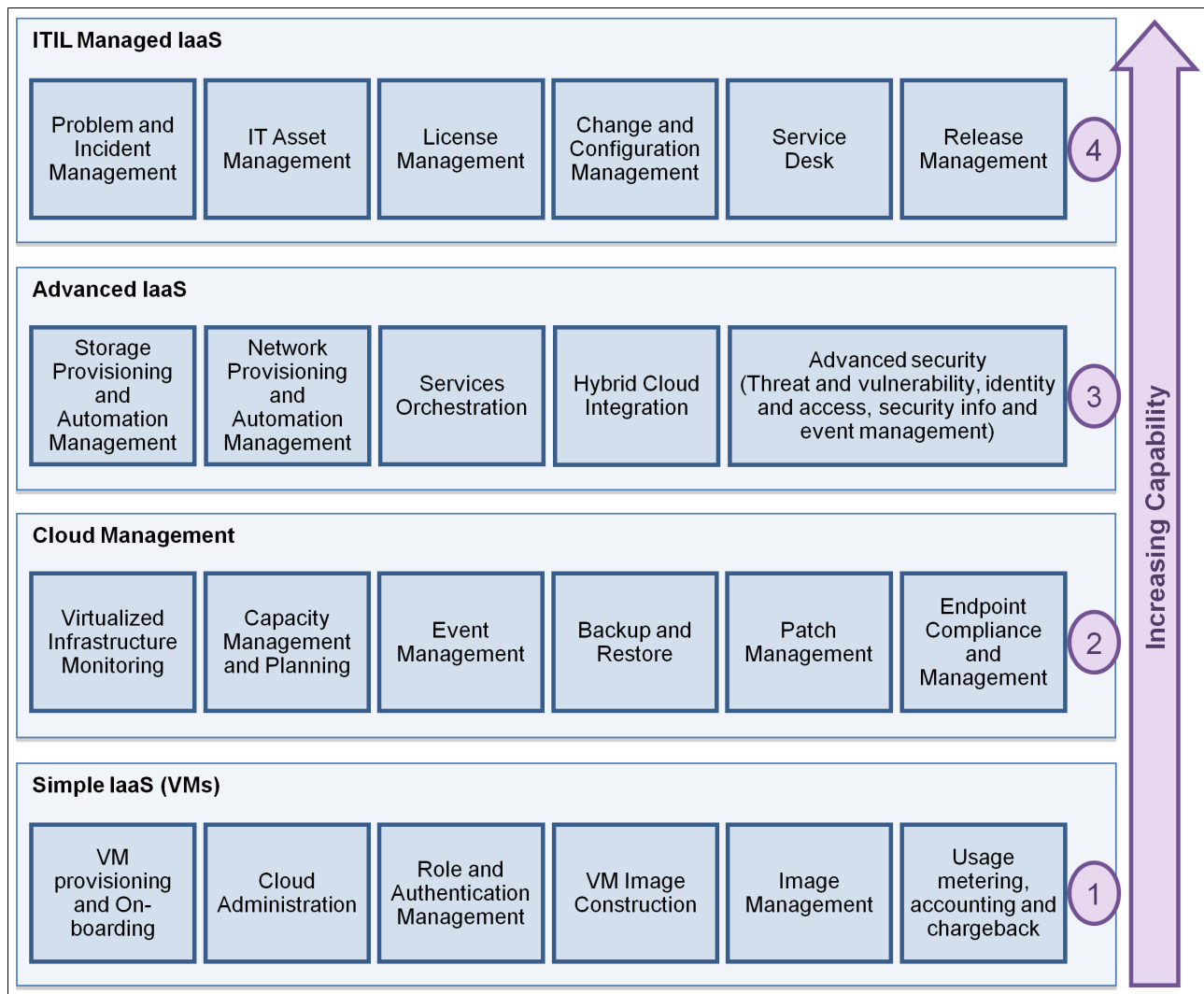


Figure 2-3 Increasing capabilities of the Cloud Enabled Data Center model

2.2 Security and the CCRA

To help you see the relationship between security and the CCRA, this section addresses the following two topics:

- ▶ 2.2.1, "Business drivers for a secure reference architecture" on page 19
- ▶ 2.2.2, "Security requirements" on page 22

2.2.1 Business drivers for a secure reference architecture

Client research has shown security concerns are the top barrier to adoption of cloud technology.

The key business drivers in the CCRA approach are to offer these benefits:

- ▶ Use cloud infrastructure with confidence that they are secure, compliant, and meet regulatory requirements.
- ▶ Leverage existing investment and extend current infrastructure to implement security for virtual infrastructure.
- ▶ Provide ease of use and automation of security steps to enable ready-to-use capabilities for cloud.
- ▶ Maintain service level compliance, accuracy, repeatability and traceability for the cloud environment.

Based on the IBM Security Framework, the CCRA articulates eight security foundational controls relating to cloud computing environments. The controls and IT processes for each foundational control are as follows:

- ▶ Cloud governance:
 - Policies for controlling the movement of tenant workloads between data-centers, countries or geographical regions.
 - Policies governing access to tenant workloads by third parties such as cloud service support vendors.
 - Governance of which cloud service providers are used and under which conditions, for example, based on the classification of data in the workload.
- ▶ Security governance, risk management and compliance:
 - Necessary policy controls are documented; control mechanisms identified.
 - Appropriate testing of controls; internal review of operation of the control.
 - Inspection of the controls by internal and external parties.
 - Reporting on controls.
 - Executive and external notification of noncompliance.
 - Capture and reporting on risk management status.
 - Auditing cloud service provider and tenant activities.
 - Providing tenants with visibility of security policies and audit events related to their tenancy, but not those relating to other tenants.
- ▶ Problem and information security incident management:
 - Problems are documented and managed according to severity.
 - Logging is maintained on critical systems.
 - Log retention period and periodic log reviews.
 - Identification and alerting of incidents.
 - Processes for response and reporting of incidents.
- ▶ Identity and access management:
 - Initial identity verification, federated identity management.
 - Complexity rules, expiration period, password reuse.

- User roles defined with access entitlements.
- System access is granted, periodically reviewed, and revoked based on business need.
- Access is logged, accountability maintained.
- Identify and resolve *separation of duties* conflicts.
- Strong authentication and encryption of remote administration.
- Monitor privileged access.
- ▶ Discover, categorize, protect data, and information assets:
 - Encrypt confidential and business-critical data at rest, for example at the application level or storage system level.
 - Encrypt confidential and business critical data in motion.
 - Management and protection for keys and certificates.
 - Managed backups are implemented, encrypted, and physically secured.
 - Inventory, periodic reconciliation, tracking of movement.
 - Expiration and data destruction; virtual system decommissioning.
 - Policies and logs are regularly reviewed.
 - Isolation of tenant-specific data at rest from other tenants, for example discretionary access control (DAC).
 - Policy-based control of access to data at rest within a tenant, for example allowing sharing or otherwise as required.
 - Isolation of tenant-specific workloads from other tenants through network segregation.
 - Protection of tenant data from cloud service provider administrators.
- ▶ Information systems acquisition, development and maintenance:
 - Documented deployment process.
 - Change control processes.
 - Administrative control and limits.
 - Vulnerability scanning.
 - Image hibernation and reactivation.
- ▶ Secure infrastructure against threats and vulnerabilities:
 - Well defined policy defined with required controls.
 - Encryption of confidential and business critical data.
 - Intrusion detection and prevention at network, host, and hypervisor levels.
 - Vulnerability detection and remediation.
- ▶ Physical and personnel security:
 - Access is granted, reviewed, and revoked.
 - Physical barriers, doors are alarmed, and alarms are tested.
 - Network cabling and infrastructure are secured and caged.
 - Visitors are escorted and logged, logs are securely retained.
 - Personnel security background checks are made.
 - Privileged access controls and monitoring personnel.

The relationship of the IBM Security Framework to the eight cloud security foundational controls is shown in Figure 2-4.

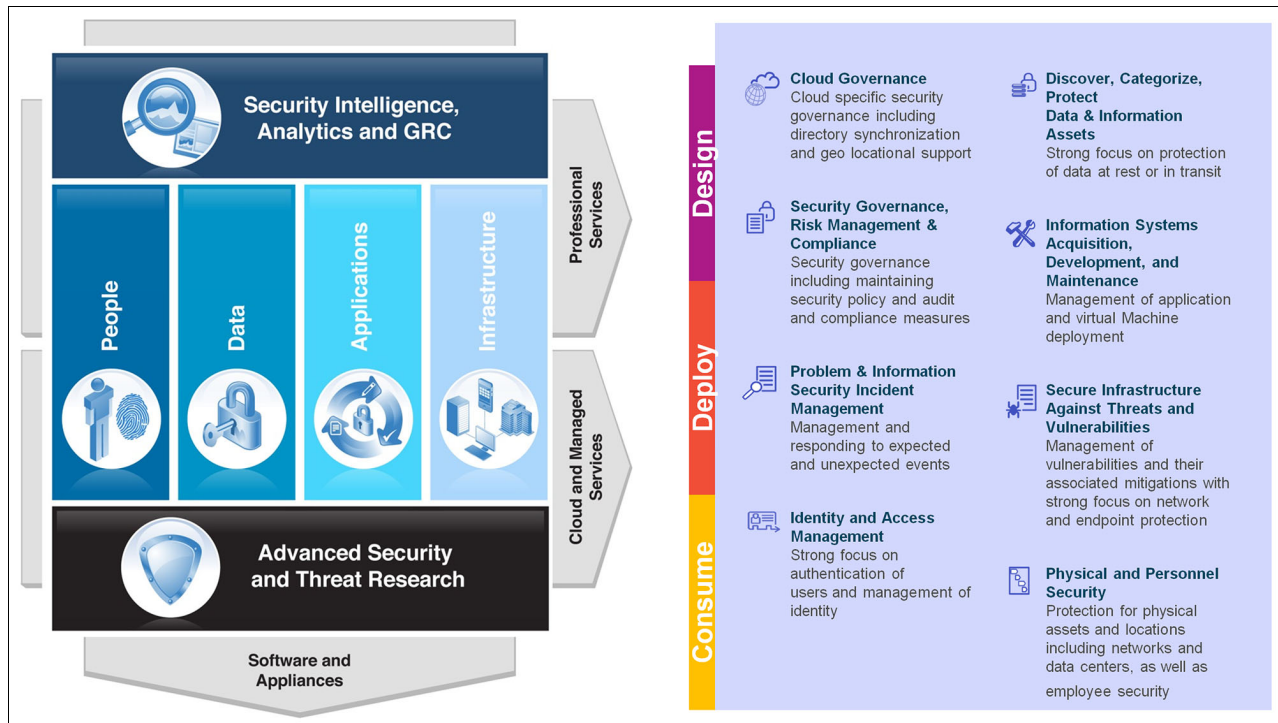


Figure 2-4 Relationship of IBM Security Framework and cloud security foundational controls

Different security controls are appropriate for different cloud consumer needs. One size does not fit all. The challenge becomes one of integration, coexistence, and recognizing what solution is best for a particular workload.

Each cloud adoption pattern offers benefits for the cloud consumer such as cutting expenses and complexity, getting speed in go-to-market, innovating business models, and getting prompt access to software ready for business solutions. Those benefits are shown in Figure 2-5.

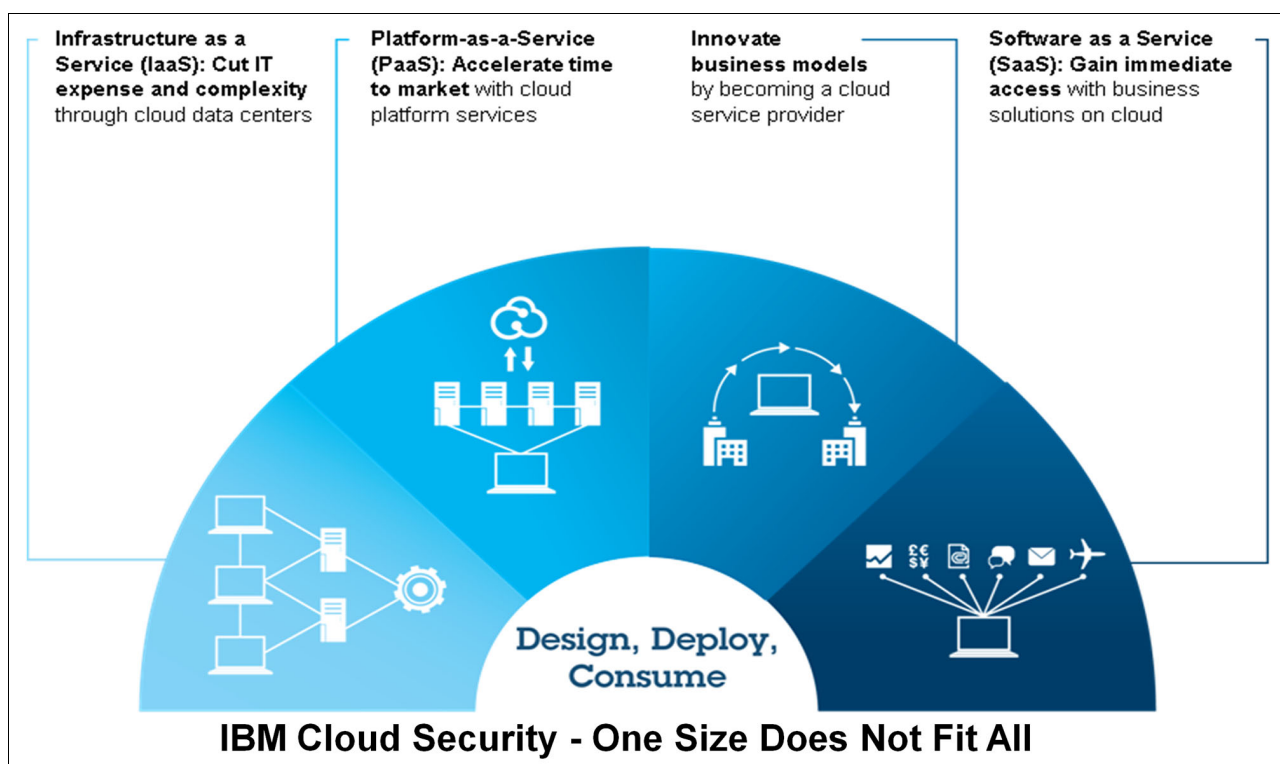


Figure 2-5 Cloud adoption patterns

For a Cloud Enabled Data Center (IaaS cloud), the CCRA identifies five key security activity areas, primarily in infrastructure and identities:

- ▶ Manage data center user identities.
- ▶ Secure virtual machines.
- ▶ Patch virtual images.
- ▶ Monitor logs on all resources.
- ▶ Isolate networks.

2.2.2 Security requirements

In the IBM Cloud Computing Reference Architecture, security, resiliency, performance, and consumability underpin all other cloud components.

Built on the IBM Security Framework, IBM security surrounding clouds focuses on developing trusted virtual domains, authentication, isolation management, policy and integrity management, and access control (Figure 2-6 on page 23), resulting in cloud environments that are secure by design.

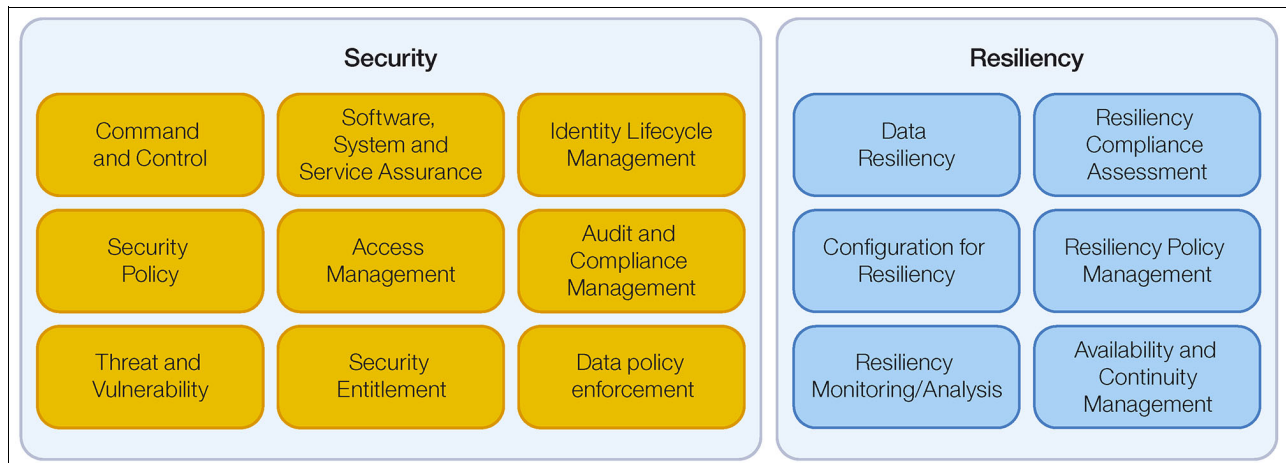


Figure 2-6 Cloud security and resiliency components

To achieve a secure Cloud Enabled Data Center, consider the following key security requirements when designing your cloud infrastructure.

- ▶ Identity and access management:
 - Manage data center identities and securely connect users to the cloud (authentication and authorization).
 - Provide role based access to cloud resources: image library, and storage.
 - Provision user IDs on the virtual machine for access to the VM or LPAR.
 - Manage confidentiality and integrity of the storage, images and metadata associated with the master image.
- ▶ Protect virtual infrastructure:
 - Secure and protect the virtual infrastructure (VM instances, LPAR, hypervisors) as per IT Security Policy.
- ▶ Provide visibility into virtual infrastructure:
 - Maintaining audit logs for virtual infrastructure compliance and audit readiness.
- ▶ Integrate with existing infrastructure and automate complex services.
 - Integrate with existing security capabilities and provide automation for identity and access management, endpoint management and log management and visibility into cloud infrastructure.

Important: Always address these security requirements as early as possible in the design of your cloud infrastructure.

Multitenancy

A private cloud can be, but does not need to be, multitenanted in its first implementation. However, do consider the possibility of a multitenanted cloud in the early design phases.

An on-premises private cloud might reduce data protection concerns but does not eliminate them. For example, isolating workloads and data between different business units to reduce conflicts of interest is a valid approach.

2.3 Cloud computing and regulatory compliance

Regulatory compliance refers to externally imposed conditions on transactions in business systems and their companies. Regulations can be imposed by industry bodies, standards organizations and government agencies.

Non-conformance with regulations can result in legal ramifications. A civil or criminal liability or regulatory penalty from a security incident can negatively affect the business.

This section offers an overview of several regulations and standards that may be applicable to your cloud computing environment.

2.3.1 Government regulations and agencies

This section describes the following regulations and agencies:

- ▶ “Federal Information Security Management Act (FISMA)” on page 24
- ▶ “National Institute of Standards and Technology (NIST)” on page 24
- ▶ “Federal Risk and Authorization Management Program (FedRAMP)” on page 25
- ▶ “Health Insurance Portability and Accountability Act (HIPAA)” on page 25
- ▶ “Sarbanes-Oxley (SOX)” on page 26

Federal Information Security Management Act (FISMA)

The Federal Information Security Management Act (FISMA) of 2002 is a federal law enacted by the United States (US) Congress. FISMA requires each federal agency to develop document and implement an agency-wide program to provide information security for the information and information systems that support the operations and assets of the agency, including those provided or managed by another agency, contractor, or other source.

For more information, see the FISMA website:

<http://www.dhs.gov/fisma>

National Institute of Standards and Technology (NIST)

The National Institute of Standards and Technology (NIST) is responsible for producing many of the information security standards and guidelines that are used by US federal agencies. In particular NIST is assigned to several specific responsibilities, including the development of these:

- ▶ Standards to be used by federal agencies to categorize information and information systems based on the objectives of providing appropriate levels of information security according to a range of risk levels.
- ▶ Guidelines recommending the types of information and information systems to be included in each category; and
- ▶ Minimum information security requirements (management, operational, and technical security controls) for information and information systems in each such category.

The NIST Guidelines on Security and Privacy in Public Cloud Computing, NIST Special Publication 800-144, indicates which documents (listed in Table 2-1 on page 25) are relevant to cloud computing, particularly for those organizations that deal with the United States Federal Government.

Important: The list of publications in Table 2-1 is not intended to be exhaustive, and cannot be considered complete for the regulations of every industry or country. More current publication revisions and updates might exist on the NIST website:

<http://csrc.nist.gov>

Table 2-1 NIST publications relevant to cloud computing

Publication	Title
FIPS 199	Standards for Security Categorization of Federal Information and Information Systems
FIPS 200	Minimum Security Requirements for Federal Information and Information Systems
SP 800-18	Guide for Developing Security Plans for Federal Information Systems
SP 800-34, Revision 1	Contingency Planning Guide for Federal Information Systems
SP 800-37, Revision 1	Guide for Applying the Risk Management Framework to Federal Information Systems
SP 800-39	Managing Information Security Risk
SP 800-53, Revision 3	Recommended Security Controls for Federal Information Systems and Organizations
SP 800-53, Appendix J	Privacy Control Catalog
SP 800-53A, Revision 1	Guide for Assessing the Security Controls in Federal Information Systems
SP 800-60	Guide for Mapping Types of Information and Information Systems to Security Categories
SP 800-61, Revision 1	Computer Security Incident Handling Guide
SP 800-64, Revision 2	Security Considerations in the System Development Life Cycle
SP 800-86	Guide to Integrating Forensic Techniques into Incident Response
SP 800-88	Guidelines for Media Sanitization
SP 800-115	Technical Guide to Information Security Testing and Assessment
SP 800-122	Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)
SP 800-137	Information Security Continuous Monitoring for Federal Information Systems and Organizations

Federal Risk and Authorization Management Program (FedRAMP)

The US Federal Risk and Authorization Management Program (FedRAMP) is a government-wide program that provides a standardized approach to security assessment, authorization, and continuous monitoring for cloud service providers. This approach uses a “do once, use many times” framework that saves cost, time, and staff required to conduct redundant agency security assessments.

For more information, see the FedRAMP website:

<http://cloud.cio.gov/fedramp>

Health Insurance Portability and Accountability Act (HIPAA)

The Health Insurance Portability and Accountability Act of 1996 (HIPAA) is a government act enacted to protect the privacy and security of individually identifiable health information. The HIPAA Security Rule sets national standards for the security of electronic protected health information.

The HIPAA mandates that all healthcare organizations effectively meet administrative, technical and physical safeguards to protect the privacy of patient information, and maintain data integrity for employees, customers and shareholders.

For more information, see the HIPAA website:

<http://www.hhs.gov/ocr/privacy>

Sarbanes-Oxley (SOX)

Following the corporate scandals of the year 2000, and a five trillion dollar dot.com crash, an overhaul of U.S. Securities Law resulted in the Sarbanes-Oxley Act of 2002 (often referred to as SOX), which has a complementary mission: “To protect investors by improving the accuracy and reliability of corporate disclosures made pursuant to the securities laws, and for other purposes.”

The Sarbanes-Oxley Act was developed to address all the complexities of investor reporting - not to mention individual accountability and integrity. Essentially, SOX requires corporations to make public the information that investors need to make informed decisions. This means that IT departments, like other business entities, must constantly look for new and innovative ways to manage and report critical corporate information.

For more information, search for Sarbanes-Oxley at the US Securities and Exchange Commission (SEC) website:

<http://www.sec.gov>

Other standards: Although the examples are based on US regulations, they might be applicable in other countries that deal with the US. Other countries, states, or regional intergovernmental organizations, such as Organisation for Economic Co-operation and Development (OECD), Asia-Pacific Economic Cooperation (APEC), or European Economic Area (EEA), might have adopted US standards, or have similar or analogous regulations.

2.3.2 Standards organizations

This section describes the following standards organizations:

- ▶ “International Organization for Standardization (ISO)” on page 26
- ▶ “Common Criteria (CC)” on page 27

International Organization for Standardization (ISO)

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) produce a family of information security standards, known as the ISO/IEC 27000 series.

Of particular relevance to cloud computing are the ISO/IEC 27001:2013 and ISO/IEC 27002:2013 documents, which deal with the specification of an information security management system and provide a code of practice for information security. Other standards also exist, such as the ISO/IEC 27018:2014 for code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors.

For more information, see the ISO 27000 series of standards website:

<http://www.27000.org>

Common Criteria (CC)

Common Criteria (CC) is an international standard for computer security certification. One of the CC main roles is to provide the widest available mutual recognition of secure IT products. CC replaced, or unified several pre-existing standards, into one widely recognized certification.

Common Criteria certifications scrutinize all security aspects of a product: design, source code, source code control, development process, and flaw remediation processes.

For more information, see the Common Criteria website:

<http://www.commoncriteriaportal.org>

2.3.3 Industry bodies

This section describes the following industry bodies:

- ▶ “Payment Card Industry Data Security Standards (PCI-DSS)” on page 27
- ▶ “Control Objectives for Information and Related Technology (COBIT)” on page 27
- ▶ “BITS” on page 28
- ▶ “American Institute of Certified Public Accountants (AICPA)” on page 28

Payment Card Industry Data Security Standards (PCI-DSS)

The Payment Card Industry (PCI) Security Standards Council offers robust and comprehensive standards and supporting materials to enhance payment card data security. These materials include a framework of specifications, tools, measurements and support resources to help organizations ensure the safe handling of cardholder information at every step.

The Payment Card Industry Data Security Standard (PCI-DSS) applies to any organization that collects, stores, processes, or transmits credit card holder data, or interacts with a third-party company that does. Achieving and maintaining PCI-DSS compliance can be costly and time-consuming, and non-compliance can result in extra fees or increased transaction charges. PCI-DSS is an internationally recognized standard.

For more information, see the PCI website:

<https://www.pcisecuritystandards.org>

Control Objectives for Information and Related Technology (COBIT)

Control Objectives for Information and Related Technology (COBIT) is a business framework for the governance and management of enterprise IT with the objective of optimizing information and technology investment and use. COBIT was created by the Information Systems Audit and Control Association (ISACA). The version COBIT 5 is based on five principles:

- ▶ Meeting stakeholder needs
- ▶ Covering the enterprise end-to-end
- ▶ Applying a single, integrated framework
- ▶ Enabling a holistic approach
- ▶ Separating governance from management

For more information, see the COBIT website:

<http://www.isaca.org/cobit>

BITS

BITS is the technology policy division of the Financial Services Roundtable (FSR). BITS addresses newly emerging threats and opportunities, particularly those related to cyber-security, fraud reduction, and critical infrastructure protection in the financial services industry.

Note: BITS was originally an acronym for Banking Industry Technology Secretariat, but that definition is no longer used.

For more information, see the BITS website:

<http://www.fsroundtable.org/bits>

American Institute of Certified Public Accountants (AICPA)

American Institute of Certified Public Accountants (AICPA) is an association representing the accounting profession, setting United States auditing standards for private companies, nonprofit organizations, and federal, state, and local governments.

Generally Accepted Privacy Principles (GAPP) are privacy principles and criteria that have been developed by AICPA and Chartered Professional Accountants (CPA) Canada to assist organizations in creating a privacy program that addresses their privacy risks and business opportunities.

Statement on Standards for Attestation Engagements (SSAE) No. 16 was drafted with the intention and purpose of updating the United States service organization reporting standard. SSAE 16 replaces Statement on Auditing Standards No. 70 (SAS 70) as the authoritative guidance for reporting on service organizations.

To learn more about the GAPP and SSAE 16, see the AICPA website and search for GAPP and SSAE 16:

<http://www.aicpa.org>

2.3.4 Summary

The cloud implementor is responsible for determining what regulations and standards are required to achieve compliance. This can vary depending on industry and location of data centers or users. Organizations dealing with the United States Federal Government can use the NIST documents (see Table 2-1 on page 25) as a starting point for example.

2.4 Security guidance

This section provides generic examples of guidance for secure systems within the CCRA security model of a cloud enabled data center (or IaaS cloud) implemented using Power Systems hardware and software. More specific guidelines are in other chapters of this book.

The following topics are covered in this section:

- ▶ 2.4.1, “Manage identities and access” on page 29
- ▶ 2.4.2, “Secure virtual machines” on page 29
- ▶ 2.4.3, “Patch default images” on page 30
- ▶ 2.4.4, “Manage logs and audit data” on page 30
- ▶ 2.4.5, “Network isolation” on page 31

2.4.1 Manage identities and access

For managing identities and access control, consider the following details:

- ▶ IBM Cloud Manager (ICM) with OpenStack, IBM Cloud Orchestrator and IBM PowerVC support Lightweight Directory Access Protocol (LDAP) as a user authentication directory. IBM Cloud Orchestrator supports OpenLDAP and IBM Directory Server. PowerVC and IBM Cloud Orchestrator support OpenLDAP and Microsoft Active Directory. Connections from the management layer server to the directory server should be encrypted with Transport Layer Security (TLS).
- ▶ PowerKVM and PowerVM both support LDAP at the operating system level. However, using LDAP authentication for the hypervisor is not critical because users will access hypervisor resources through other management layers (for example, PowerVC or ICM).
- ▶ Role-based access should be configured for PowerVC and IBM Cloud Manager. *Admin*, *deployer*, and *viewer* are the three default LDAP roles or user groups used in PowerVC. Keystone is the OpenStack component that handles users identities. IBM Cloud Manager also supports user roles and groups membership through the OpenStack user and tenant model provided by the Keystone component.
- ▶ Management layers that have a web interface (for example, HMC, Kimchi, PowerVC, and ICM) should have HTTPS enabled for user access. Where it is possible to choose which cipher-suite is used in SSL handshake negotiation, the strongest suite supported by the users' browsers should be selected. For example, TLS 1.2 might be mandated by some security standards, but might not be supported by all browsers. The authentication process in particular, must be encrypted from end-to-end (that is, from user browser to LDAP server). In addition, consider replacing all self-signed certificates, with certificates from a recognized internal or external certificate authority (CA).
- ▶ An *on-premises* private hybrid cloud required that consideration is given to federated identity management technologies such as OpenID, OAuth, and SAML.
- ▶ Manage privileged administrator user access through processes or tools. Access to privileged user accounts such as *root* or *padmin* should be managed securely to prevent accountability and compliance issues, and decrease the risk of sabotage and data loss.

2.4.2 Secure virtual machines

For securing virtual machines, observe the following details:

- ▶ Encryption of data at rest might be a requirement of some security standards (for example, PCI-DSS or HIPAA). The hypervisor can support encrypted data at the SAN level by use of a disk array that supports encryption (for example, IBM System Storage® DS8000®) or an encrypting SAN switch (for example, IBM System Storage SAN32B-E4). An alternative solution is to use an encrypted file system on the guest operating system. Consider these examples:
 - On AIX: Encrypted File System (EFS) and IBM Spectrum™ Scale
 - On IBM i: Encrypted ASP Enablement (IBM i option 45)
 - On Linux: dm-crypt, LUKS, or eCryptFS
- ▶ Encryption of data backups is also a factor to consider. Tape drives with encryption hardware do not impose any performance degradation. Key management also becomes a priority.

- ▶ Packet filtering at the virtual machine host level should be implemented. On AIX, TCP/IP filters are included in the IPSec packages (`bos.net.ipsec.keymgt` and `bos.net.ipsec.rte`). On IBM i, you can use IBM i Navigator to create a Packet Rules, which is a combination of network address translation and IP filtering. The default packet filter for recent Linux distributions is `iptables`. Some time must be invested to survey what TCP/IP ports must be allowed through the packet filter, based on administration, management, and application requirements. Access should be restricted to the appropriate IP addresses or subnets wherever possible.
- ▶ Hosts should also be subject to *hardening* by removing unnecessary software packages. For example, is an HTTP server package required on a compute node? This is best done at the image library level, but there are advantages and disadvantages with *multi-purpose* images. A better approach is to start with a hardened base image, and require the user to choose packages or functionality from a software catalog at deployment time.
- ▶ Follow standard system administrator recommendations. A non-exhaustive list can include: password policy enforcement (complexity, aging, reuse), changing of default system and application passwords, backups of critical data, use of network time synchronization protocols (NTP) to synchronize time across systems. Most of these practices are mandated by regulations such as PCI-DSS and HIPAA, so are not considered optional.
- ▶ Routine vulnerability scanning is also suggested. Scanning can be done on the local host or remotely over the network.
- ▶ Virtual machines and physical hardware must go through a secure disposal process to prevent the inadvertent leakage of confidential data after redeployment of virtual or physical resources.

2.4.3 Patch default images

For patching default images, observe the following details:

- ▶ Cloud administrators must be diligent in ensuring their image library is kept current with regards to security patches. More sophisticated cloud management applications will have more advanced tools for maintaining image libraries.
- ▶ Compliance tools such as endpoint managers or management frameworks can be beneficial in keeping both virtual machines up to date with security patches, and also hypervisors and management servers.

2.4.4 Manage logs and audit data

For managing logs and audit data, observe the following details:

- ▶ All layers of the Power Systems stack support syslog for logging of system events. All management nodes and virtual machines should be configured to forward syslog messages to a event management system at the least. The use of a Security Information and Event Manager (SIEM) is suggested, particularly one with advanced threat protection (ATP).
- ▶ In addition to syslog, SIEM systems can have event or audit agents that are required to be installed on the endpoints (management servers or virtual machines).
- ▶ Time must be invested to produce a baseline of events processed by the event manager with minimal false positives. Procedures must be in place to act on escalated events in a timely manner.

2.4.5 Network isolation

For managing network isolation, observe the following details:

- ▶ Management interfaces (HMC, hypervisors) must be kept isolated from networks used for virtual machines and other networks.
- ▶ In addition, networks used for functions like live migration and backups must not be accessible by the user.
- ▶ Use the appropriate combination of virtual networks (VLANs), physical network separation, and firewalls as deemed appropriate. Again, the use of next generation firewalls with integrated intrusion prevention systems (IPS) and advanced threat protection (ATP) is suggested.
- ▶ With increasing uptake of hybrid clouds, virtual private networks (VPNs) also becomes a requirement. Two-factor authentication is usually a requirement for VPN access.

2.5 Usage scenarios

This section illustrates the security process in this book with the following usage scenarios:

- ▶ 2.5.1, “Generic use case with cloud-enabled data center” on page 31
- ▶ 2.5.2, “Typical PowerKVM use case” on page 32
- ▶ 2.5.3, “Typical PowerVM use case” on page 33

2.5.1 Generic use case with cloud-enabled data center

The generic use case in a cloud-enabled data center or IaaS cloud is shown in Figure 2-7.

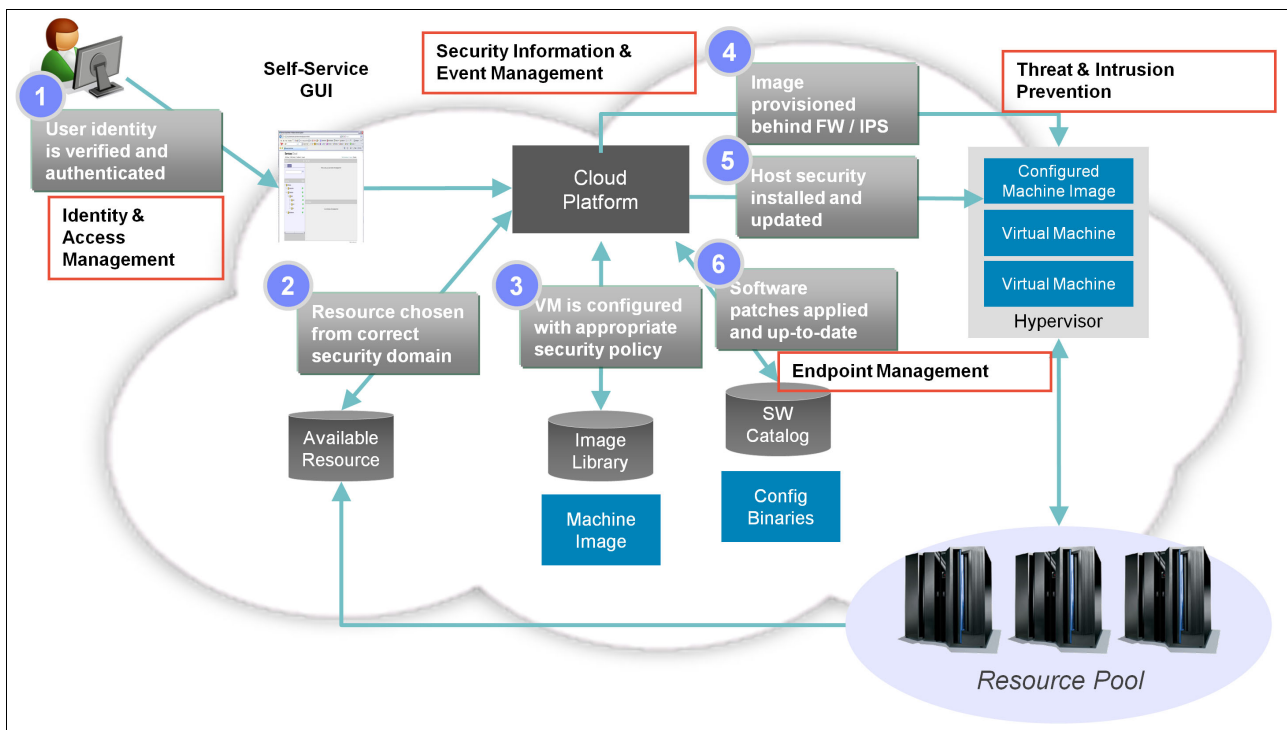


Figure 2-7 Generic use case for Cloud Enabled Data Center

The following security enforcement points are numbered in Figure 2-7 on page 31:

1. The user logs in to a self-service graphical interface by using a web browser. Security considerations include these items:
 - Encrypted HTTPS connection with appropriate key lengths and ciphers used.
 - Use of an LDAP user directory.
 - LDAP authentication should be encrypted end-to-end.
 - Identity lifecycle should be managed in an enterprise wide manner.
2. When authenticated, the user should be granted access to appropriate resources based their functional role. Additional access controls can be implemented with enterprise-wide access management tools.
3. A required virtual machine image is selected from a pre-configured image library. Stored images in the image library should have all appropriate management tools installed and required security policy applied, for example, endpoint management agents, password policy, and so on.
4. Provision the image onto a dedicated virtual machine network. Segregation of different functional networks, for example management networks, functional team networks, and so on, should be enforced with the appropriate combinations of virtual networks (VLANs), physical network separation, firewalls, and access control lists (ACLs). The use of next generation firewalls with Intrusion Prevention Systems (IPS) and Advanced Threat Protection (ATP) is suggested.
5. Host security of the virtual machine should be installed or updated. This might include updates to endpoint manager policies, configuring logging destinations for event management, or installation of specific users keys for access to the virtual machine.
6. There might be too many combinations of application software versions to allow preconfigured images of every combination. Software would then be installed and updated from a software catalog post-installation.

Specific examples for Power Systems components are discussed in subsequent chapters of this book.

2.5.2 Typical PowerKVM use case

The PowerKVM hypervisor is ideally suited to organizations that have few or no existing Power Systems, but want to take advantage of the enterprise virtualization and I/O bandwidth features of the POWER8 scale-out systems. The organization has a requirement only for Linux workloads.

This type of organization might place high importance on open source software (such as Linux and KVM) and has experienced Linux users or administrators. The Linux based administration of the PowerKVM hypervisor is simpler because a dedicated HMC appliance is not required.

Advanced Virtualization Management or Cloud Management is available through the PowerVC, Cloud Manager with OpenStack, or IBM Cloud Orchestrator products.

The security recommendations for the PowerKVM hypervisor are discussed in Chapter 5, “IBM PowerKVM security” on page 87.

2.5.3 Typical PowerVM use case

The PowerVM hypervisor is the ideal choice for organizations with existing experience with Power Systems. These organizations have the requirement to run IBM AIX or IBM i workloads and also Linux on Power workloads. These organizations are able to leverage their existing pre POWER8 Power Systems, and Hardware Management Consoles, while taking advantage of the performance advantages of the POWER8 Power Systems.

Currently the following PowerVM features are used by PowerVC:

- ▶ Dynamic LPARs
- ▶ NPIV
- ▶ vSCSI storage
- ▶ Shared storage pools
- ▶ Shared processors
- ▶ Capping of LPARs
- ▶ Multiple shared processor pools
- ▶ Remote restart

Other PowerVM and PowerKVM features are able to be tolerated but not explicitly set.

Again, Advanced Virtualization Management or Cloud Management is available through the PowerVC, Cloud Manager with OpenStack, or IBM Cloud Orchestrator products.

The security recommendations for the PowerKVM hypervisor is discussed in Chapter 4, “IBM PowerVM security” on page 61.

2.6 Integration with IBM software

The security processes and preferences described in this book integrate with the following types of IBM products:

- ▶ Security Information and Event Management (SIEM)
- ▶ Identity and access management
- ▶ Endpoint management
- ▶ Threat and intrusion prevention

For more information about IBM security products, see the IBM Security website:

http://www.ibm.com/security/products/?lnk=sec_home

2.6.1 Security Information and Event Management (SIEM)

IBM provides the following SIEM solutions:

- ▶ IBM QRadar® Log Manager

IBM QRadar Log Manager is a comprehensive solution for organizations that want to implement a distributed event log manager to collect, archive, and analyze network and security event logs. QRadar Log Manager is used to perform an integrated analysis of network and security information of the cloud infrastructure and provide awareness of network security threats that need to be resolved quickly.

For more information, see the IBM Security QRadar Log Manager website:

<http://www.ibm.com/software/products/en/qradar-log-manager>

► IBM SmartCloud® Monitoring

IBM SmartCloud Monitoring monitors the health and performance of a private cloud infrastructure, including environments that contain physical and virtualized components. This software provides the tools that are needed to assess current health and capacity and model expansion, as needed. IBM SmartCloud Monitoring provides the following capabilities:

- Visibility into the cloud infrastructure, including environments that contain physical and virtualized components
- Monitoring of heterogeneous environments for visibility and control into all areas of the infrastructure, such as physical, virtual, and cloud
- Policy-driven analytics for intelligent workload placement
- What-if capacity planning to accommodate capacity growth while optimizing the usage of the existing environment

For more information, see the IBM SmartCloud Monitoring website:

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

2.6.2 Identity and access management

IBM provides the following identity and access management solutions:

► IBM Security Identity Manager

Security Identity Manager is a solution that delivers security rich, policy-based user and role management across the IT infrastructure. This solution is used to map the Cloud Enabled Data Center roles to the appropriate users in the organization and to automate the creation, modification, recertification, and termination of user privileges throughout the user lifecycle.

For more information, see the IBM Security Identity Manager website:

<http://www.ibm.com/software/products/en/identity-manager>

► IBM Security Access Manager for Web

IBM Security Access Manager for Web is an appliance-based security solution that provides both access control and protection from web-based threats, including the top 10 web application risks identified by the Open Web Application Security Project (OWASP).

The appliance combines the capabilities of a reverse proxy server, single sign-on server, web application firewall, centralized policy server, load balancer and distributed session caching, packaged in one solution that helps secure web access to applications, while also protecting them from the latest threats. Highly scalable and configurable, the appliance is designed to help organizations reduce the costs and complexities of multi-channel access management.

For more information, see the following websites:

- IBM Knowledge Center; search for IBM Security Access Manager for Web:

<https://www.ibm.com/support/knowledgecenter/?lang=en>

- IBM Security Access Manager:

<http://www.ibm.com/software/products/en/access-mgr>

2.6.3 Endpoint management

IBM provides the following endpoint management solution:

- ▶ IBM BigFix®

IBM BigFix consolidates common management tasks into a modular, multi-platform solution that delivers real-time visibility and control over all endpoints, regardless of the operating system, physical location or bandwidth.

For more information, see the following websites:

- IBM BigFix “Getting Started” topic in the IBM Knowledge Center:

<http://ibm.co/1nNERcG>

- IBM Security BigFix website:

<http://www.ibm.com/security/bigfix/>

2.6.4 Threat and intrusion prevention

IBM provides the following threat and intrusion prevention solutions:

- ▶ IBM Security Network Intrusion Prevention System

IBM Security Network Intrusion Prevention solutions provide comprehensive protection and reduce the cost and complexity that are associated with deploying and managing endpoint solutions. Security Network Intrusion Prevention is used to discover threats to the Cloud Enabled Data Center infrastructure. It is also used to take preventive measures against threats to these high value assets and to protect the workloads from threats such as SQL injection and cross-site scripting attacks.

For more information, see the IBM Security Network Intrusion Prevention System website:

<http://www.ibm.com/software/products/en/network-ips>

- ▶ IBM Security SiteProtector™ System

IBM Security SiteProtector System is a centralized management system that unifies management and analysis for network, server, and endpoint security agents and appliances. It reduces the cost and complexity of security management, helps you monitor and measure your exposure to vulnerabilities and demonstrate regulatory compliance. IBM Security SiteProtector system can help minimize your overall risk and increase the efficacy of your security team, while optimizing cost efficiency.

IBM Security SiteProtector is supported by and integrated with the IBM X-Force® research and development team, its tools, online security information and security updates.

For more information, see the IBM Security SiteProtector System website:

<http://www.ibm.com/software/products/en/site-protector-system>

2.7 Conclusion

For a cloud environment to meet management and user expectations, in terms of performance, value, and security, an essential task is for all of the following items to be adequately handled in the cloud design phase:

- ▶ Security
- ▶ Privacy
- ▶ Resiliency
- ▶ Service management
- ▶ Governance

The IBM Cloud Computing Reference Architecture (CCRA) delivers best practices in a standard and methodical way. The CCRA is secure by design, and supports several cloud computing service models (IaaS, PaaS, SaaS, and CSP) and service delivery models (public, private, and hybrid).

This chapter focuses on the security aspects of an infrastructure as a service (IaaS) cloud running on IBM Power Systems, and some of the government or industry regulations and standards that might apply in this model.



Part 2

Power cloud components

This part provides security considerations about the components that can compose a cloud-based infrastructure solution based on IBM Power Systems.

This part focuses on the components described in the following chapters:

- ▶ Chapter 3, “IBM Hardware Management Console (HMC) security” on page 39
- ▶ Chapter 4, “IBM PowerVM security” on page 61
- ▶ Chapter 5, “IBM PowerKVM security” on page 87
- ▶ Chapter 6, “IBM PowerVC security” on page 137
- ▶ Chapter 7, “IBM Cloud Manager with OpenStack security” on page 169
- ▶ Chapter 8, “IBM Bluemix secure gateway” on page 191



IBM Hardware Management Console (HMC) security

This chapter discusses security features of the IBM Hardware Management Console (HMC) to help you understand HMC security capabilities.

The following topics are covered in this chapter:

- ▶ 3.1, “Introduction to the HMC” on page 40
- ▶ 3.2, “User interfaces” on page 40
- ▶ 3.3, “Network interfaces” on page 41
- ▶ 3.4, “User and role management” on page 43
- ▶ 3.5, “Monitoring and auditing HMC access” on page 50
- ▶ 3.6, “Security enhancements and compliance” on page 52
- ▶ 3.7, “HMC and security zones” on page 56
- ▶ 3.8, “Conclusion” on page 60

3.1 Introduction to the HMC

The Hardware Management Console (HMC) is a Linux-based appliance used for configuration, management, and maintenance of IBM Power Systems servers.

The HMC is specifically built to manage IBM Power Systems servers. With the HMC, you can perform the following operations:

- ▶ Define, deploy, manage, and maintain logical partitions.
- ▶ Open virtual terminals to logical partitions.
- ▶ Manage and maintain IBM Power System servers.
- ▶ Manage PowerVM features, including virtualization features.
- ▶ Manage IBM Power System server firmware.
- ▶ Manage Capacity On Demand resources.
- ▶ Define a focal point for service and maintenance operations.

A single HMC can manage multiple IBM Power Systems servers. IBM Power Systems servers can be connected to a second HMC for redundancy. In a redundant configuration, either of the two HMCs can be used to perform any tasks that are specific to HMC. Redundant HMCs are independent from each other and can be used interchangeably; no primary and secondary designation exists.

Use of redundant HMCs: A preferred practice is to use redundant HMCs to manage your IBM Power Systems environment.

3.2 User interfaces

HMC operations can be performed using either a web-based graphical user interface (GUI) or a command-line interface (CLI). After HMC users log in and are authenticated, they are confined to a restricted environment.

HMC users of the web-based GUI have access only to functions that are provided by the GUI.

HMC users of the CLI must first log in and authenticate, after which they are placed in a restricted Bash shell. The purpose of the restricted shell is to confine HMC users and protect the integrity of the entire HMC environment.

The HMC is a Linux-based appliance and not a general-use Linux system. Installing any other Linux-compatible application on the HMC is *not* allowed.

Commands are executed within the restricted shell and are limited to operations required to configure, manage, or service the managed systems or the HMC itself. HMC users have access only to HMC-specific commands.

The commands that users can run are subject to limitations enforced by the role that is assigned to the user by the HMC administrator.

Users are not allowed to run the `su` command. The only alternative to escape from the restricted shell and obtain access to a root shell is to request a password from IBM Technical Support. The password can allow users to access a shell from where they can use the `su` command to root.

User interface security

The HMC is used to configure and manage complex environments that can include a considerable number of IBM Power Systems. Because HMC users have access to a powerful set of functions, such as powering on and off logical partitions, powering off managed systems, and performing dynamic LPAR (DLPAR) operations, user interface security becomes paramount. The HMC provides these interfaces:

- ▶ Console

This interface allows connectivity for HMC users that use a local console that is attached to the HMC. Users must authenticate with user ID and password. The restricted shell is the only choice, even for users who log in by using the HMC console.

- ▶ Web-based graphical user interface (GUI)

The GUI allows connectivity for HMC users using a browser. Users must authenticate with user ID and password. Users are authorized to perform only the tasks that are included in the role assigned. Additionally, the user must be explicitly granted remote access by the HMC administrator. Communication between browser and the HMC is secured by SSL on the web server that is running on HMC.

- ▶ Command-line interface (CLI)

The CLI allows connectivity for HMC users through SSH. Users must authenticate with user ID and password. Users must be explicitly granted remote access and are confined to restricted shell.

3.3 Network interfaces

IBM Power Systems that are managed by HMC require Ethernet connectivity between the HMC and the server flexible service processor (FSP).

PowerVM features such as DLPAR, Live Partition Mobility (LPM), or IBM Active Memory™ Sharing (AMS) require TCP/IP connectivity between LPARs and the HMC.

Tip: Given the importance of connectivity between the HMC and the FSP, a preferred practice is to have a completely separated network that contains one HMC interface and the FSPs of all managed systems. Also a preferred practice is to have this network be isolated. The HMC network interface connected to this network is called *private* and the HMC considers the network private.

By default, the FSP uses DHCP to obtain its IP address. On private networks, the HMC can be configured to act as a DHCP server for a predefined set of non-routable IP address ranges. This allows the HMC to control the assignment of IP addresses and prevent IP address conflicts. The HMC also supports IPv6 protocol.

All other HMC network interfaces are considered *open*. Open network usually refers to any general, public network that contains devices other than HMC private interfaces and FSPs.

The HMC can have multiple network interfaces and can be connected to either private or public networks.

Alternatively, the FSPs can be assigned dedicated IP addresses or, although not recommended, even connected to open networks.

The HMC has a built-in firewall feature that filters network traffic and a predefined list of ports that can be open. The HMC does not allow IP forwarding. The HMC GUI allows adding and removing ports from of list of ports currently open as shown in Figure 3-1.

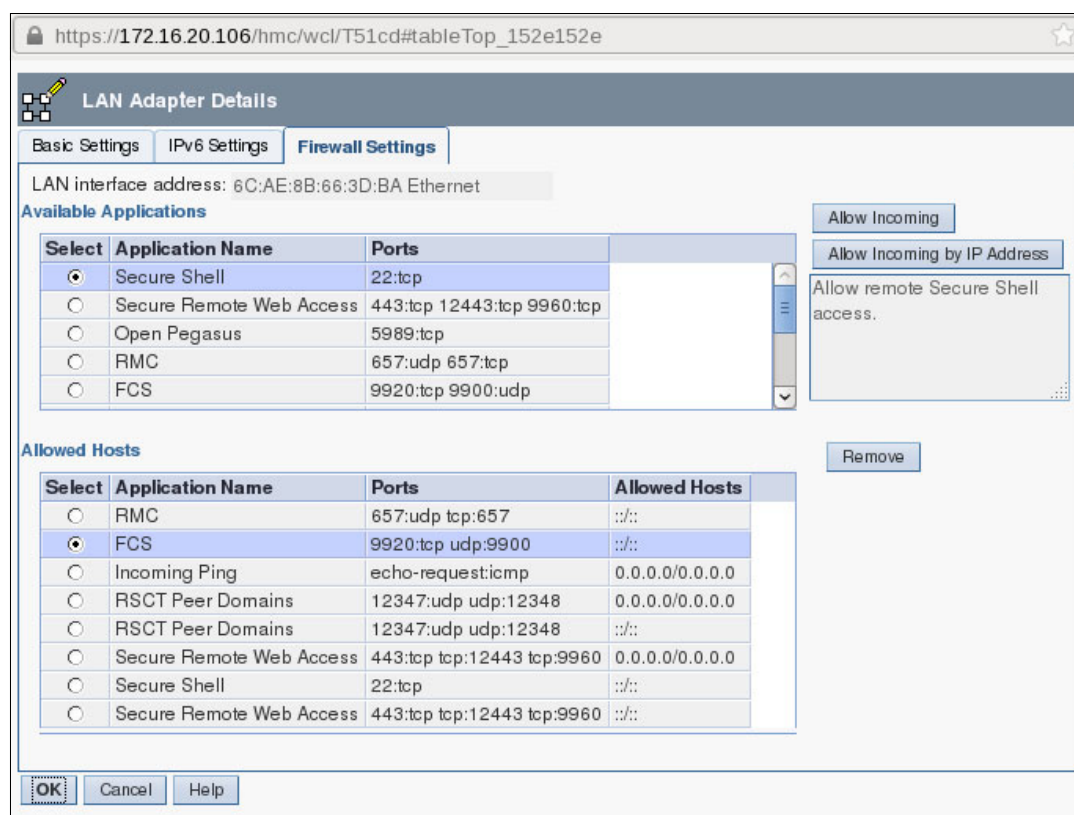


Figure 3-1 Using the HMC GUI to manage open ports on the HMC firewall

The HMC administrator can restrict the accessibility of HMC network services to individual network interfaces and also filter what systems are allowed access to individual services. The default firewall policy is *Deny all*.

Connectivity to the flexible service processor (FSP)

Communication between HMC and FSP uses TCP/IP. The FSP uses a 2048-bit RSA certificate. The algorithm used to sign the certificate is SHA256. The certificate is used to establish an SSL connection between HMC and FSP. The certificate is not alterable on either endpoint of the communication. For extra security, the FSP can be configured to accept connections only from a specific set of IP addresses.

The FSP is managed through the Advanced System Management Interface (ASMI). The ASMI can be accessed using a web browser, an ASCII terminal, or through the HTTPS proxy feature of the HMC. The FSP has a firewall configured that allows access to the following ports:

- ▶ 443 (HTTPS)
- ▶ 30000 (HMC Command Interface)
- ▶ 30001 (Stream interface for Virtual TTY support)

After the connectivity between HMC and FSP is established, all operations performed by HMC must be authenticated. The process of user authentication consists of two levels of identification and authorization as follows:

1. The user must first authenticate with an HMC user ID that has HMC administrative privileges.
2. The user must then authenticate with a valid user ID and password for the FSP.

The ASMI allows for a list of predefined users as shown in Table 3-1.

Table 3-1 ASMI user IDs

User ID	Default password	Authority level
general	general	General user
admin	admin	Administrator
celogin	Contact IBM for password	Authorized service provider
celogin1	Not set, disabled by default	Authorized service provider
celogin2	Not set, disabled by default	Authorized service provider
dev	Contact IBM for password	Developer user, service only

Note: Managed systems store configuration information in the NVRAM and can operate even in the absence of the HMC. However, changing partition configurations requires the HMC.

3.4 User and role management

Because HMC users have access to a powerful set of functions such as powering on and off logical partitions, powering off managed systems, and performing DLPAR operations, you must fully understand the user identities and privileges that are handled by the HMC.

This section discusses the following topics:

- ▶ 3.4.1, “Users” on page 43
- ▶ 3.4.2, “Roles” on page 44
- ▶ 3.4.3, “Practical scenario of using users and customized roles” on page 45

3.4.1 Users

The HMC is a stand-alone Linux-based appliance and has its own user space that is not related to any LPAR user space or the user space of another HMC.

HMC authentication can be done by using one of the following ways:

- ▶ Local authentication
Users authenticate with a user ID and password.
- ▶ Kerberos authentication
Users authenticate using Kerberos. This method requires the configuration of a Kerberos Domain Controller (KDC) server. Users defined to use Kerberos authentication use this method even when they log in to HMC locally.

- ▶ LDAP authentication

Users authenticate using LDAP. This method requires the configuration of an LDAP server. Users defined to use LDAP authentication use this method even when they log in to HMC locally.

Note: The HMC has a predefined user ID named *hscroot*, and the default password is *abc123*. This user ID cannot be deleted.

3.4.2 Roles

Upon creation, HMC users must be assigned *roles* by the HMC administrator. The HMC role-based access control ensures that only authorized users that have explicitly granted privileges can perform specific tasks. Each role consists of a set of granular privileges. By assigning roles, users are confined to perform only a limited set of individual operations.

HMC allows the definition of task roles and managed resource roles. Individual HMC users are assigned one task role and one or more managed resource roles.

Task roles

The set of individual operations a user can perform represents a *task role*. For example, an individual operation can be powering on a managed system or opening a console to an LPAR.

The HMC has the following predefined task roles:

- ▶ hmcserVICerep
- ▶ hmcviewer
- ▶ hmcoperator
- ▶ hmcpe
- ▶ hmcSuperadmin

By default, the actions HMC users are allowed can be executed on all resources that are managed by the HMC for which the actions are applicable. For example, a user who is assigned a task role that allows that user to power off a managed system, can perform this operation on all systems that the HMC manages.

Managed resource roles

The set of objects on which an HMC user can perform the operations allowed by its task role is called a *managed resource role*. By default, the HMC has a single predefined managed resource role named *AllSystemResources*, which includes all resources that are managed by the HMC.

Access control to HMC can be further refined by using *customized* roles. Customized roles can be created by copying and modifying existing predefined roles. Customized task roles restrict the set of allowed operations. Customized resource roles limit the scope of allowed operations to a specific set of objects.

The HMC administrator can confine individual users to a limited set of operations by defining customized roles that include fine-grained privileges.

The HMC has several predefined user roles as listed in Table 3-2.

Table 3-2 Predefined HMC user roles

Role name	Privileges	Description
hmcoperator	Operator	The Operator is responsible for daily system operation.
hmcsuperadmin	Super Administrator	The Super Administrator acts as the root user of the manager of the HMC. The Super Administrator has unrestricted authority to access and modify most of the HMC system.
hmcpe	Product Engineer	A Product Engineer helps in support operations, but cannot access HMC user management functions. To provide support access for your system, you must create the administrator user IDs with the Product Engineer role.
hmcservicerep	Service Representative	A Service Representative is an employee who is at your location to install, configure, or repair the system.
hmcviewer	Viewer	A Viewer can view HMC information, but cannot change any configuration information.

Based on the predefined roles, the HMC administrator can define customized roles. Customized roles can be refined to such an extent to include atomic privileges to individual objects. For example, users can be granted privileges so that they are limited to powering on and off an individual LPAR.

3.4.3 Practical scenario of using users and customized roles

The following steps illustrate a practical example of defining customized roles:

1. List the existing task roles using the **lsaccfg** HMC command (Example 3-1). This example has no customized task roles; the command displays only predefined roles. HMC predefined task roles cannot be deleted.

Example 3-1 Using HMC *lsaccfg* command to list existing task roles

```
hscroot@hmc7:~>lsaccfg -t taskrole -F name,parent
hmcservicerep,Predefined
hmcviewer,Predefined
hmcoperator,Predefined
hmcpe,Predefined
hmcsuperadmin,Predefined
```

- Using the GUI, create a customized task role, named LPAR_operator, based on the predefined role hmcsuperadmin. Assign only the individual privileges of activating and shutting down the partitions (Figure 3-2).

Note: HMC has suggestive predefined names for all individual privileges that can be assigned to customized roles. Even if a customized task role initially includes a large set of privileges, it can be refined and limited to include only the intended privileges of the task role.

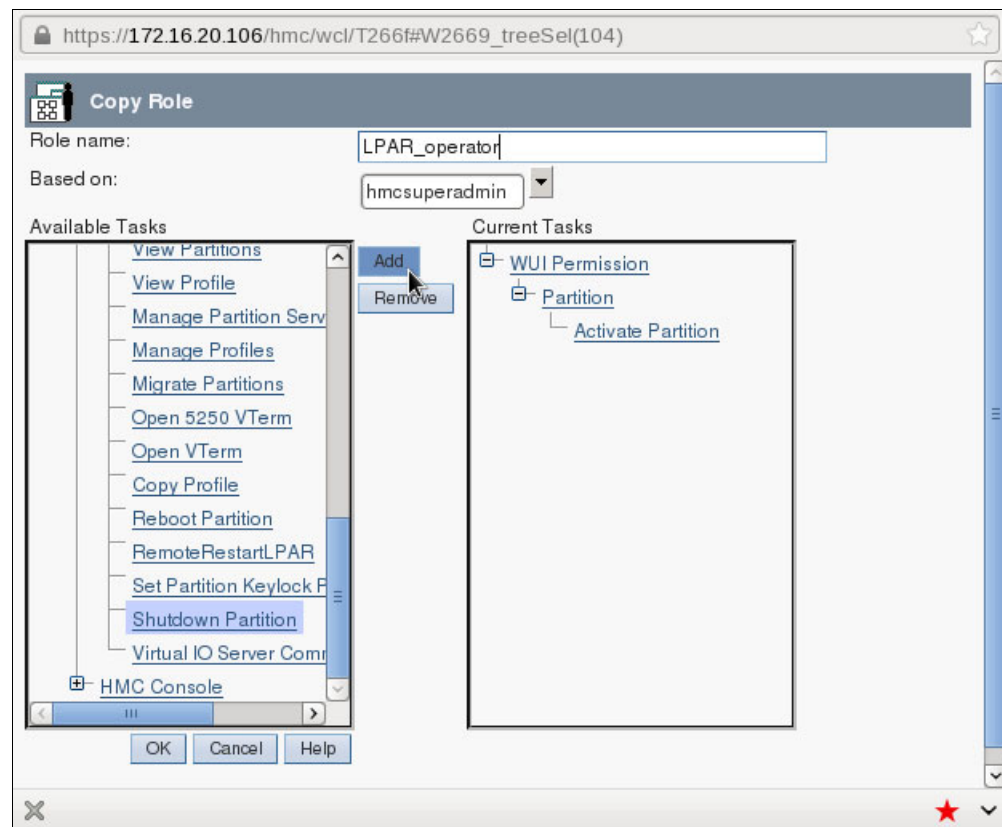


Figure 3-2 Adding individual privileges to a customized task role

- With the CLI, verify the newly defined task role by using the `lsaccfg` HMC command (Example 3-2).

Example 3-2 Listing individual privileges associated to a task role

```
hscroot@hmc7:~>lsaccfg -t taskrole --filter "taskroles=LPAR_operator"
name=LPAR_operator,parent=hmcsuperadmin,resources=lp:ActivateLPAR+ShutdownLPAR
```

4. With the HMC GUI, create two customized managed resource roles as follows:
- Create the first role. In this example it is called LPARs_on_system_p750_0 and includes all LPARs defined on system p750_0 (Figure 3-3).

Note the existence of convenient predefined subsets of manageable objects such as all logical partitions or individual frames.

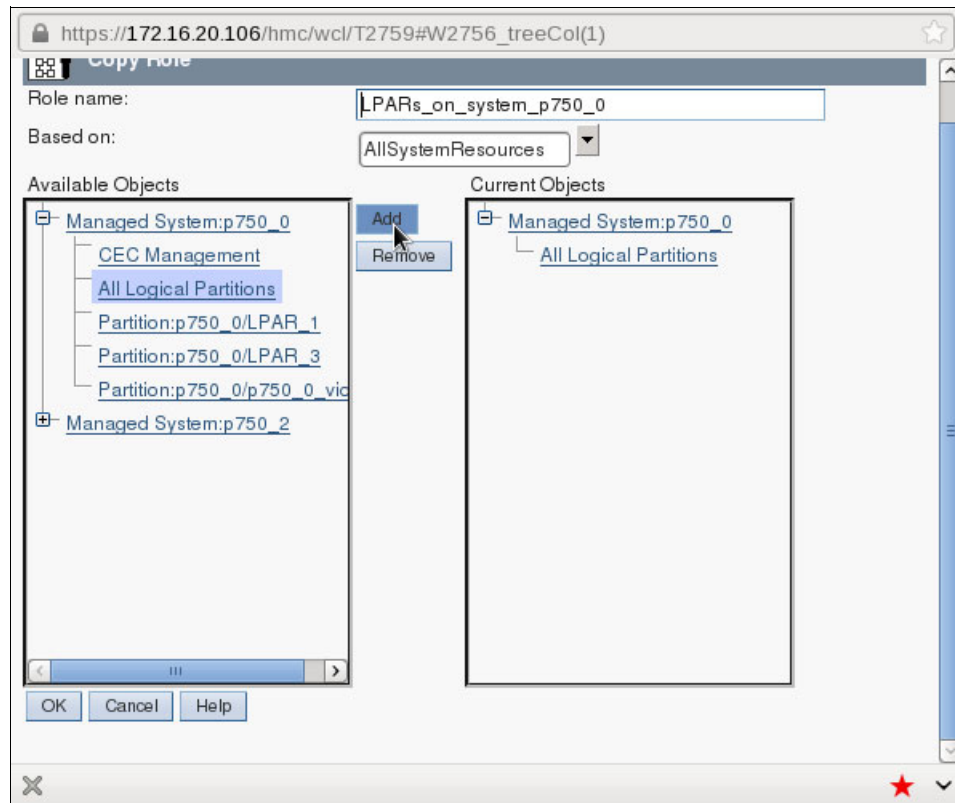


Figure 3-3 Assigning all existing LPARs to a managed resource role

- b. Create the second role. In this example it is named `Even_LPArS_on_p750_2` and includes two LPARs defined on system `p750_2` (Figure 3-4).

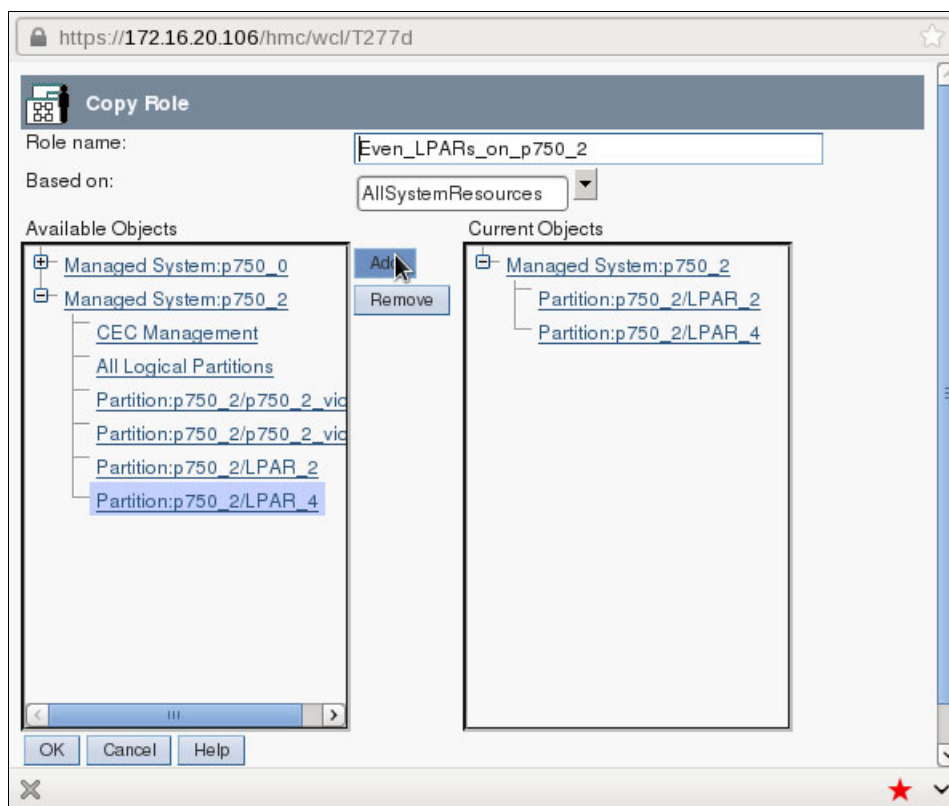


Figure 3-4 Assigning individual LPARs to a managed resource role

5. With the CLI, verify the newly defined resource role by using the `lsaccfg` HMC command (Example 3-3).

Example 3-3 Listing resources assigned to a managed resource role

```
hscroot@hmc7:~>lsaccfg -t resourcerole
name=LPArS_on_system_p750_0,resources=lpAr:root/ibmhscS1_0|ALL_PARTITIONS*8233-E8B*10DD51P|IBMHSC_Partition
name=Even_LPArS_on_p750_2,"resources=lpAr:root/ibmhscS1_0|4*8233-E8B*061AB2P|IBMHSC_Partition,lpAr:root/ibm
hscS1_0|6*8233-E8B*061AB2P|IBMHSC_Partition"
```

6. With the HMC GUI, create two HMC users (a user can be assigned multiple managed resource roles):
 - a. Create user `lpar_operator`. This user is assigned the task role `LPAR_operator` and the managed role `Even_LPArS_on_p750_2`.
 - b. Create user `frame_operator`. It is assigned the task role `LPAR_operator` and the managed role `LPArS_on_system_p750_0` (Figure 3-5).

Figure 3-5 Creating a user and assigning task and managed resource roles

7. With the CLI, verify the newly defined HMC users by using the `lshmcusr` HMC command (Example 3-4).

Example 3-4 Listing HMC users and roles associated to them

```
hscroot@hmc7:~>lshmcusr --filter "taskroles=LPAR_operator" -F
name:resourcerole:description
lpar_operator:Even_LPArS_on_p750_2:LPAR operator
frame_operator:LPArS_on_system_p750_0:Frame Operator
```

8. With the CLI, show how user `lpar_operator` can perform only the activities included in this user's role (Example 3-5).

Example 3-5 HMC user can only perform the tasks allowed by user roles

```
lpar_operator@hmc7:~>lsyscfg -r sys
HSC350B The user does not have the appropriate authority.

lpar_operator@hmc7:~>lsyscfg -r lpar -m p750_2 -F LPAR_2
HSC350B The user does not have the appropriate authority.

lpar_operator@hmc7:~>chsysstate -r lpar -b sms -o on -m p750_2 -n LPAR_2 -f default

par_operator@hmc7:~>lsrefcode -r lpar -m p750_2 -F "lpar_name refcode time_stamp"
--filter "lpar_names=LPAR_2"
HSC350B The user does not have the appropriate authority.

lpar_operator@hmc7:~>chsysstate -r lpar -o shutdown -m p750_2 -n p750_2_vio1
The partition entered was not found. Please check your entry and retry the command.

lpar_operator@hmc7:~>chsysstate -r lpar -o shutdown -m p750_2 -n LPAR_4
HSC05DF The partition is not in a state under which this operation can be performed.
Check the state of the partition.
```

Consider the following information shown in Example 3-5:

- The `lpar_operator` cannot get general information about the environment. Use the `lsyscfg` HMC command to get more information regarding the system that is prohibited.
- The `lpar_operator` cannot get information even for the partitions that this user is allowed to power on and off. Use the `lsyscfg` HMC command to get more information regarding the system that is prohibited.
- The `lpar_operator` is authorized to power on partition `LPAR_2` in SMS mode. The privilege `Activate_LPAR` allows the user to run the `chsysstate` HMC command with appropriate flags.
- The `lpar_operator` is not authorized to see the reference code while the partition is starting.
- The `lpar_operator` is not authorized to perform any operation on other LPARs. The attempt to shutdown the partition called `p750_2_vio1` fails.
- The `lpar_operator` is authorized to receive error messages related to the tasks allowed. Shutting down partition `LPAR_4` is a legitimate operation, and running the `chsysstate` HMC command is allowed. However, the command fails for a technical reason and the appropriate error message is displayed to the `lpar_operator`.

3.5 Monitoring and auditing HMC access

The HMC is designed so that access to the HMC and all operations performed from the HMC can be monitored, logged, and audited. This section examines the following details:

- ▶ 3.5.1, “Access monitoring” on page 51
- ▶ 3.5.2, “Access auditing” on page 51

3.5.1 Access monitoring

The **lslogon** HMC command can be used to display users who are connected to the HMC through the CLI or web-based GUI. The command can also be used to monitor the tasks performed by the users as shown in Example 3-6.

Example 3-6 Using HMC lslogon command to monitor HMC access

```
hscroot@hmc7:~>lslogon -r ssh -u
user_name=hscroot,TTY_id=pts/0,logon_time=2014-08-19 14:08,access_location=172.16.254.26
user_name=lp_ar_operator,TTY_id=pts/2,logon_time=2014-08-19 14:33,access_location=172.16.254.26

hscroot@hmc7:~>lslogon -r webui -u
user_name=hscroot,session_id=4,logon_time=07/25/2014 15:35:55
user_name=hscroot,session_id=20,logon_time=08/19/2014 14:38:17

hscroot@hmc7:~>lslogon -r webui -t
task_id=107,task_name=Open Terminal Window,session_id=4,start_time=08/19/2014 13:53:35,user_name=hscroot
task_id=120,task_name=Manage Task and Resource Roles,session_id=20,start_time=08/19/2014
15:53:21,user_name=hscroot
```

Consider the following information shown in Example 3-6:

- ▶ HMC users hscroot and lp_ar_operator are logged on to the HMC through SSH and use the restricted shell.
- ▶ The time when SSH connections were initiated is displayed for both users.
- ▶ The IP address from where the connections were initiated are displayed for both users.
- ▶ Two web sessions are opened by hscroot user (session ID 4 and ID 20). The time when sessions were initiated is displayed for both sessions.
- ▶ Two tasks are currently run by hscroot (task ID 107 and ID 120). Task 120 is used to manage HMC tasks and resource roles.

3.5.2 Access auditing

HMC activity is carefully recorded in system logs. Events that occurred on the HMC can be divided in two classes:

Hardware These events relate to hardware problems that occur on the HMC.

Console These events log the operations performed from the HMC.

HMC events can be listed by using the **lssvcevents** HMC command. Operations such as creating users and roles and assigning privileges are recorded in the HMC logs.

Example 3-7 shows how to list all console events that occurred during the last 40 minutes.

Example 3-7 Using HMC commands to list HMC events

```
hscroot@hmc7:~>lssvcevents -t console -i 40
time=08/19/2014 01:38:38,text=HSCE2103 User name hscroot: Changed property assign of user lp_ar_operator to
Even_LPARs_on_p750_2.
time=08/19/2014 01:38:38,text=HSCE2298 UserName hscroot: User Assignment redefined for User lp_ar_operator.
time=08/19/2014 01:38:37,text=HSCE2101 User name hscroot: Created user lp_ar_operator with role
LPAR_operator.
time=08/19/2014 01:38:37,text=HSCE2103 User name hscroot: Changed property pwage of user lp_ar_operator to
99999.
time=08/19/2014 01:35:22,text=HSCE2354 User name hscroot: New user frame_operator copied from hscroot with
task role [LPAR_operator].
```

```
time=08/19/2014 01:35:21,text=HSCE2103 User name hscroot: Changed property assign of user frame_operator to LPARs_on_system_p750_0.
time=08/19/2014 01:35:21,text=HSCE2298 UserName hscroot: User Assignment redefined for User frame_operator.
time=08/19/2014 01:35:21,text=HSCE2101 User name hscroot: Created user frame_operator with role LPAR_operator.
time=08/19/2014 01:35:21,text=HSCE2103 User name hscroot: Changed property pwage of user frame_operator to 99999.
time=08/19/2014 01:27:06,text=HSCE2314 UserName hscroot: Group Even_LPARs_on_p750_2 created.
time=08/19/2014 01:21:52,text=HSCE2314 UserName hscroot: Group LPARs_on_system_p750_0 created.
time=08/19/2014 01:07:01,text=HSCE2302 UserName hscroot: New Role LPAR_operator created with hmcsuperadmin as parent role.
```

3.6 Security enhancements and compliance

This section discusses the following HMC security enhancements and compliance:

- ▶ 3.6.1, “Security compliance” on page 52
- ▶ 3.6.2, “HMC security enhancements” on page 52
- ▶ 3.6.3, “Data replication” on page 55
- ▶ 3.6.4, “Customizing HMC encryption” on page 55

3.6.1 Security compliance

The National Institute of Standards and Technology (NIST) Special Publications 800-131a, NIST 800-131a, and FIPS-140-2, among others, require the use of new ciphers, better random number generators, and the use of Transport Layer Security (TLS) version 1.2 in Secure Sockets Layer (SSL) communication. NIST requirements deprecated some algorithms such as Data Encryption Standard (DES) algorithm for ciphers. SP800-131a strengthens security by defining which algorithms can be used and their minimum strengths.

To support the SP 800-131a standards, the following enhancements were made:

- ▶ Upgrade JVM to a new version which provides NIST support.
- ▶ Enable TLS 1.2 and be prepared to disable protocols less than TLS 1.2.
- ▶ Cryptographic keys adhere to a minimum key strength of 112 bits.
- ▶ Digital signatures are a minimum of SHA2.
- ▶ Use approved random number generator (Java Only).

3.6.2 HMC security enhancements

Starting with HMC version 7.7.0, the HMC introduces several enhancements to support security features required by compliance with NIST requirements. HMC commands are extended to support the transition to the security levels that are required by NIST 800-131a.

Example 3-8 on page 53 shows how to use HMC commands to verify the effects of enabling HMC compliance with NIST requirements.

Example 3-8 Verifying and enabling HMC compliance with NIST 800-131a requirements

```
hscroot@hmc7:~>lshmc -r -F security
legacy
```

```
hscroot@hmc7:~>lshmcencr -c webui -t a
"avail_encryptions=TLS_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_DHE_
RSA_WITH_AES_256_CBC_SHA,TLS_ECDH_RSA_WITH_AES_256_CBC_SHA,TLS_DHE_DSS_WITH_AES_256_CBC_SHA
,TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_
128_CBC_SHA,TLS_DHE_RSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_ECDH_RS
A_WITH_AES_128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,TLS_ECDH_ECDSA_WITH_AES_128_CBC
_SHA,TLS_DHE_DSS_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_3DES_EDE_CBC_SHA,TLS_ECDHE_RSA_WITH_3DES
_EDE_CBC_SHA,TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA,TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA,TLS_RSA_F
IPS_WITH_3DES_EDE_CBC_SHA,TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA,TLS_DHE_DSS_WITH_3DES_EDE_C
BC_SHA,TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA,SSL_RSA_WITH_AES_128_CBC_SHA,SSL_DHE_RSA_WITH_A
ES_128_CBC_SHA,SSL_DHE_DSS_WITH_AES_128_CBC_SHA,SSL_RSA_WITH_3DES_EDE_CBC_SHA,SSL_RSA_FIPS_
WITH_3DES_EDE_CBC_SHA,SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA,SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA,S
SL_RSA_EXPORT_WITH_DES40_CBC_SHA,SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA,SSL_RSA_WITH_DES_CBC
_SHA,SSL_RSA_FIPS_WITH_DES_CBC_SHA,SSL_DHE_RSA_WITH_DES_CBC_SHA,SSL_DHE_DSS_WITH_DES_CBC_SH
A,SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA"
```

```
hscroot@hmc7:~>lshmcusr -F name:password_encryption
root:sha512
lpar_operator:sha512
frame_operator:sha512
hscroot:sha512
```

```
hscroot@hmc7:~>chhmc -c security -s modify --mode nist_sp800_131a
```

The Hardware Management Console will automatically be restarted after the security mode is changed. Are you sure you want to change the security mode (0 = no, 1 = yes)?

1

Broadcast message from root@hmc7 (Wed Aug 27 15:33:39 2014):

The system is shutting down for reboot now.

```
hscroot@hmc7:~>
```

Broadcast message from root@hmc7
(unknown) at 15:34 ...

The system is going down for reboot NOW!

Write failed: Broken pipe

```
[liviuc7420477041 ~]$ ssh hscroot@172.16.20.106
```

```
hscroot@172.16.20.106's password:
```

Last login: Wed Aug 27 15:29:55 2014 from 172.16.254.26

```
hscroot@hmc7:~>lshmc -r -F security
```

```
nist_sp800_131a
```

```
hscroot@hmc7:~>lshmcusr -F name:password_encryption
```

```
root:sha512
```

```
lpar_operator:sha512
```

```
frame_operator:sha512
```

```
hscroot:sha512
```

```
hscroot@hmc7:~>lshmcencr -c webui -t a
```

```
"avail_encryptions=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_GCM_SHA384,TL
S_DHE_RSA_WITH_AES_256_GCM_SHA384,TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_128_
GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_
RSA_WITH_AES_256_CBC_SHA256,TLS_DHE_RSA_WITH_AES_256_CBC_SHA256,TLS_RSA_WITH_AES_128_CBC_SH
A256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_DHE_RSA_WITH_AES_128_CBC_SHA256"
```

Consider the following information shown in Example 3-8 on page 53:

- ▶ The HMC runs by default in legacy mode as shown by the **1shmc** HMC command.
- ▶ Use the **1shmcencr** HMC command to list the encryptions available for the web user interface.
- ▶ Use the **1shmcsur** HMC command to list password encryption for HMC users. All user passwords use SHA512.
- ▶ User hscroot uses the **chhmc** HMC command to change the security mode. Changing the security mode requires an HMC reboot.
- ▶ Following the reboot, the HMC runs in `nist_sp800_131a` mode.
- ▶ Use the **1shmcsur** HMC command to list the password encryption for HMC users. All user passwords use SHA512.
- ▶ Use the **1shmcencr** HMC command to list the updated list of web user interface encryptions available in the new security mode.

When enabling `nist_sp800_131a` mode, you must ensure that your SSH client or web browser supports the new requirements and you are still able to connect the HMC.

For example, you must verify your web browser version and, depending on the result, you might be required to enable individual browser features or change specific browser settings. If your browser does not support the new security requirements, you typically get an SSL handshake error similar to the one shown in Figure 3-6.

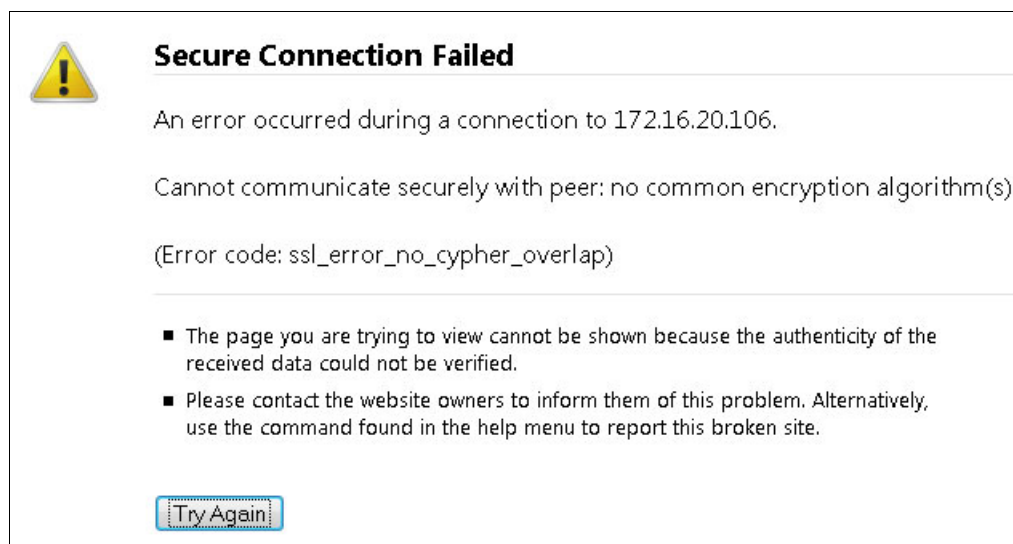


Figure 3-6 Browser SSL handshake failure

Table 3-3 describes the HMC browser requirements after NIST SP800-131A support is enabled on the HMC.

Table 3-3 HMC browser requirements after NIST SP800-131A support is enabled

Browser Name	Browser version	NIST (TLS v1.2) supported
Firefox	<ul style="list-style-type: none"> ▶ 1-18 ▶ ESR 10,17 ▶ 19-23 	No
	<ul style="list-style-type: none"> ▶ 24-26 ▶ ESR 24 	Yes, disabled by default
	<ul style="list-style-type: none"> ▶ 27+ ▶ ESR31 and later 	Yes
Internet Explorer	6,7	No
	8 and later	Yes, disabled by default
	11	Yes
Chrome	0-29	No
	30 and later	Yes

When running in legacy mode, the HMC uses an RSA certificate with a key size of 512. The signature algorithm used to sign the certificate is MD5. When enabling `nist_sp800_131a` mode, the HMC uses a certificate with RSA key size of 2048 bits and SHA256 to sign the certificate.

3.6.3 Data replication

In cloud and multitenant environments, the HMC data replication feature allows multiple HMCs to share data. With the Customizable Data Replication service you can configure a set of HMCs to automatically replicate certain types of data. Data that can be synchronized automatically includes user profile data, Kerberos configuration data, LDAP configuration data, and password policy configuration data.

Communication between multiple HMCs uses legacy Chatlet communication via port 9920. Enabling `nist_sp800_131a` mode automatically disables legacy mode. Legacy mode can be enabled by using the following HMC command:

```
chhmc -c legacyhmccomm -s enable
```

3.6.4 Customizing HMC encryption

You can customize the HMC encryption by restricting the encryptions you want to use in your environment. The HMC maintains the following lists of encryptions:

- Available** This list includes all available encryptions. By default, all available encryptions are active and can be used.
- Current** This list represents a subset of the available encryptions and includes encryptions that are active and permitted to use.

Example 3-9 shows how the list of available encryptions can be restricted.

Example 3-9 Restricting the list of active encryptions

```
hscroot@hmc7:~>lshmcencr -c passwd -t c
curr_encryption=sha512

hscroot@hmc7:~>lshmcencr -c webui -t a
"avail_encryptions=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_DHE_RSA_WITH_AES_256_GCM_SHA384,TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA256,TLS_DHE_RSA_WITH_AES_256_CBC_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_DHE_RSA_WITH_AES_128_CBC_SHA256"

hscroot@hmc7:~>chhmcencr -c webui -o r -e
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_DHE_RSA_WITH_AES_256_GCM_SHA384,TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA256,TLS_DHE_RSA_WITH_AES_256_CBC_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

hscroot@hmc7:~>lshmcencr -c webui -t c
curr_encryptions=TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
```

Consider the following information shown in Example 3-9:

- ▶ Use the **lshmcencr** HMC command to list the set of current available encryptions for the user password. The current list includes only sha512.
- ▶ Use the **lshmcencr** HMC command to list the set of currently available encryptions for the web user interface.
- ▶ Use the **chhmcencr** HMC command to remove all available encryptions except TLS_DHE_RSA_WITH_AES_128_CBC_SHA256.
- ▶ Use the **lshmcencr** HMC command to confirm that the list of current encryptions includes only the desired options.

3.7 HMC and security zones

With IBM Power Systems, the IBM POWER® Hypervisor provides connectivity between different logical partitions using virtual Ethernet adapters. The IBM POWER Hypervisor™ provides a software implementation of an Ethernet switch that is compliant with standard IEEE 802.1Q. This software switch allows operating systems running in logical partitions to communicate using standard networking protocols.

In cloud and multitenant environments with multiple IBM Power Systems, grouping logical partitions in different *security zones* based on their communication needs is possible. Multiple logical partitions that need to communicate can be placed in the same security zone. On the contrary, logical partitions that must be isolated can be placed in different security zones. Logical partition isolation using security zones can be achieved by creating *multiple* virtual switches and configuring logical partitions with virtual Ethernet adapters to connect to a specific virtual switch.

The following topics are discussed in this section:

- ▶ 3.7.1, “Virtual switches” on page 57
- ▶ 3.7.2, “Enforcement of ACLs on virtual switches” on page 59
- ▶ 3.7.3, “ACL support for LPM” on page 59

3.7.1 Virtual switches

By default, all IBM Power Systems have an internal virtual switch named *ETHERNET0* to which all logical partitions can connect. However, defining multiple virtual switches is possible. When multiple virtual switches exist, all logical partitions are allowed by default to access all virtual switches.

The HMC provides the means to both define multiple virtual switches and to specify which logical partitions can connect to each virtual switch. Access control to virtual switches is enforced using virtual switch Access Control Lists (ACL).

The HMC task role *hmcsuperadmin* includes a task named *Manage Virtual Network*, which controls the access to IBM Power System virtual switches. The privileges included in this task allows you to create, update, and delete virtual switches and their corresponding attributes. Only HMC users that have these privileges can manage virtual switch ACLs.

The Manage Virtual Network task is shown in Figure 3-7.

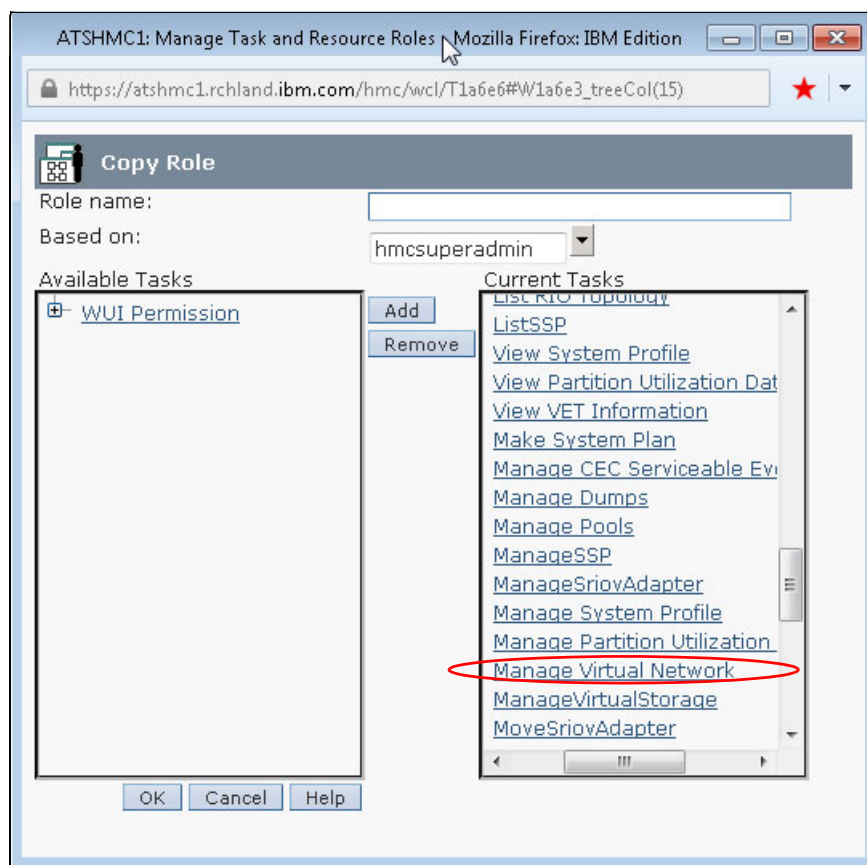


Figure 3-7 Manage Virtual Network task required to manage virtual switches access control lists

The HMC task role `hmcsuperadmin` can be used to define a customized role that includes only this individual privilege as discussed in 3.4.3, “Practical scenario of using users and customized roles” on page 45. This role can later be assigned to HMC users that are entitled to manage virtual switches.

In complex cloud and multitenant environments, you can use this technique to enforce the separation of duties by taking the following steps:

1. Ensure that all HMC users entitled to manage virtual switches are assigned a customized role that includes only this privilege.
2. Ensure that all HMC users that are not entitled to manage virtual switches are assigned a customized role that explicitly excludes this privilege.

Example 3-10 shows how to use HMC commands to create multiple virtual switches and configure existing logical partitions to use them.

Example 3-10 Creating virtual switches and configuring logical partitions to use virtual switches

```
Liviur@ATSHMC1:~> lssyscfg -r sys -m ATS_740D -F vswitch_lpar_access_list_capable
1
Liviur@ATSHMC1:~> chhwres -m ATS_740D -r virtualio --rsubtype vswitch -o a --vswitch
switch1

Liviur@ATSHMC1:~> chhwres -m ATS_740D -r virtualio --rsubtype vswitch -o a --vswitch
switch2

Liviur@ATSHMC1:~> lshwres -m ATS_740D -r virtualio --rsubtype vswitch
vswitch=ETHERNET0(Default),"vlan_ids=1,101,20",switch_mode=VEB
vswitch=switch2,vlan_ids=none,switch_mode=VEB
vswitch=switch1,vlan_ids=none,switch_mode=VEB

Liviur@ATSHMC1:~> lshwres -m ATS_740D -r virtualio --rsubtype vswitch -F
vswitch,allowed_lpar_ids
ETHERNET0(Default),all
switch2,all
switch1,all

Liviur@ATSHMC1:~> chhwres -m ATS_740D -r virtualio --rsubtype vswitch -o s --vswitch
switch1 -a "allowed_lpar_ids=11"

Liviur@ATSHMC1:~> chhwres -m ATS_740D -r virtualio --rsubtype vswitch -o s --vswitch
switch2 -a "allowed_lpar_ids=10"

Liviur@ATSHMC1:~> lshwres -m ATS_740D -r virtualio --rsubtype vswitch -F
vswitch,allowed_lpar_ids
ETHERNET0(Default),all
switch2,10
switch1,11
```

Consider the following information shown in Example 3-10:

- ▶ Use the `lssyscfg` HMC command to check if the system supports virtual switch ACLs.
- ▶ Use the `chhwres` HMC command to create a virtual switch called `switch1`. This virtual switch is used to connect logical partitions with odd IDs.
- ▶ Use the `chhwres` HMC command to create a virtual switch called `switch2`. This virtual switch is used to connect logical partitions with even IDs.
- ▶ Use the `lshwres` HMC command to confirm both virtual switches have been successfully created.

- ▶ Use the `lshwres` HMC command to verify which logical partitions can connect to the new virtual switches. All logical partitions can connect to all virtual switches.
- ▶ Use the `chhwres` HMC command to restrict the list of logical partitions that can connect to `switch1`. This example defines an ACL for `switch1` which includes only the logical partition with ID 11.
- ▶ Use the `chhwres` HMC command to restrict the list of logical partitions that can connect to `switch2`. This example defines an ACL for `switch2` which includes only the logical partition with ID 10.
- ▶ Use the `lshwres` HMC command to display the existing ACLs for both virtual switches.

3.7.2 Enforcement of ACLs on virtual switches

The POWER Hypervisor is responsible for management of all virtual switches and all ACLs associated to them. The enforcement of ACLs for virtual switches is done by the POWER Hypervisor during the following events:

- ▶ Activation of a logical partition.
- ▶ Adding (DLPAR) a virtual Ethernet adapter to running logical partition.
- ▶ Performing an LPM operation on a logical partition.
- ▶ Resuming a logical partition.

Any attempt to perform an operation that is not authorized by the ACL results in an error similar to the one shown in Example 3-11.

Example 3-11 Error message during operations not authorized by ACLs

```
HSCL02C4 The operation has failed because the virtual Ethernet adapter in slot 4
is not configured to connect to a virtual switch to which this partition is not
authorized to connect.
```

The enforcement of ACLs for virtual switches is *not* done by the PowerVM Hypervisor during the following events:

- ▶ Creation or modification of a logical partition profile
Authorized users can create or modify profiles that include virtual Ethernet adapters configured to access any virtual switch. Actual enforcement occurs upon activation of the logical partition profile.
- ▶ Validation of an LPM operation
Authorized users are not warned of any failure during an LPM validation. Actual enforcement occurs upon actual LPM of the logical partition.
- ▶ Validation of resuming operation
Authorized users are not warned of failure during resume validation. Actual enforcement occurs upon resuming of the logical partition.

3.7.3 ACL support for LPM

In cloud and multitenant environments with multiple IBM Power Systems and multiple security zones, the possibility exists to migrate a logical partition from one managed system to another and keep the logical partition within the *same security zone*.

ACLs are enforced during LPM operations. The following considerations apply:

- ▶ Virtual switch ACLs must be configured on the destination managed system *prior to* migration.
- ▶ A virtual switch with the same name is used on the destination managed system. If the switch does not exist, it is created but *not connected* to any networks.
- ▶ The partition ID is maintained for the migrating partition *if possible*. If the partition ID is already in use on the destination managed system, the first available partition ID (starting with ID 1) is used.

3.8 Conclusion

The HMC is a secure Linux-based appliance that is used to deploy and manage IBM Power Systems. The following list indicates some of the main security characteristics of the HMC:

- ▶ Secure web and command-line interfaces
- ▶ Restricted shell and a specific set of commands
- ▶ Network interfaces protected by a built-in firewall
- ▶ User space with role-based access control
- ▶ Advanced monitoring and auditing capabilities
- ▶ Compliance with current security standards
- ▶ Support for advanced security algorithms and technologies



IBM PowerVM security

IBM PowerVM is the industry-leading virtualization solution for IBM AIX, IBM i, and Linux environments based on IBM POWER technology. It provides a state-of-the-art secure virtualization solution for cloud and multitenant environments based on IBM Power Systems.

IBM PowerVM allows for system consolidation while maintaining the same degree of separation provided by the physical systems.

This chapter introduces several core concepts used by IBM PowerVM and explains how they relate to the security of IBM Power Systems environments. The content of this chapter is solely intended to help you gain a better understanding of how various features available on IBM Power Systems relate to and rely on PowerVM. This chapter explains why and how security in such environments is a logical consequence of PowerVM design and features. Note, however, that presenting a view of the internal details of PowerVM is beyond the purpose of this publication.

The following topics are covered in this chapter:

- ▶ 4.1, “IBM PowerVM overview” on page 62
- ▶ 4.2, “Isolation requirements for logical partitions” on page 62
- ▶ 4.3, “Domains of IBM Power processor cores” on page 63
- ▶ 4.4, “Processor core access modes” on page 65
- ▶ 4.5, “POWER Hypervisor” on page 65
- ▶ 4.6, “Memory isolation” on page 69
- ▶ 4.7, “I/O isolation” on page 73
- ▶ 4.8, “Logical partitions (LPARs)” on page 75
- ▶ 4.9, “Virtualization of I/O devices” on page 76
- ▶ 4.10, “Security of DLPAR operations” on page 79
- ▶ 4.11, “IBM PowerVM security management with PowerSC” on page 80
- ▶ 4.12, “Secure Logical Partition Mobility” on page 81
- ▶ 4.13, “PowerVM NovaLink” on page 85
- ▶ 4.14, “Conclusion” on page 86

4.1 IBM PowerVM overview

IBM PowerVM is a combination of hardware enablement and value-added software. It features leading technologies such as IBM POWER Hypervisor, IBM Micro-Partitioning®, Dynamic Logical Partitioning, shared processor pools, shared storage pools, Live Partition Mobility (LPM), active memory sharing, N_PortID virtualization, and suspend/resume.

Consolidation of separate physical systems onto virtualized systems relies on abstraction, virtualization, and secure sharing of existing physical resources between multiple logical partitions.

With IBM PowerVM, configuring logical partitions to use either *shared* or *dedicated* resources is possible. When using dedicated resources, physical cores, memory, and I/O devices are explicitly allocated to individual partitions so that no resources are shared by multiple logical partitions. The remainder of this chapter focuses on the security of shared resources.

For a comprehensive presentation of IBM PowerVM capabilities, see the following Redbooks publications:

- ▶ *IBM PowerVM Virtualization Introduction and Configuration*, SG24-7940
- ▶ *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590

Note: IBM PowerVM is certified at the Evaluated Assurance Level 4+. At the time of writing this publication, no vulnerabilities are reported against IBM PowerVM by either the US-CERT or by the MITRE Corporation^a.

a. <https://cve.mitre.org>

4.2 Isolation requirements for logical partitions

When running in multitenant or cloud environments, logical partitions must be completely separated and isolated. Logical partitions must not be able to get access to or tamper with resources they are not entitled to. With IBM PowerVM technology, logical partitions are not even aware of the existence of other logical partitions.

This section describes how general requirements of logical partition isolation translate into specific requirements for various types of resources such as processor cores, memory, and I/O.

4.2.1 Workload isolation

In terms of workload isolation, the following objectives must be met:

- ▶ A program, a process, a thread, or the operating system running within a logical partition *must not be capable to* affect or influence another logical partition.
- ▶ Failure of a program, a process, a thread, or the logical partition operating system itself *must not be capable to* affect or influence another logical partition.
- ▶ Resource usage rate of a logical partition *must not be capable to* affect or influence performance of another logical partition.
- ▶ A program, a process, a thread, or the operating system running within a logical partition *must not be aware of* the existence of other programs, processes, threads, or operating systems running within other logical partitions.

4.2.2 Processor core isolation

In terms of processor core isolation, the following objectives must be met:

- ▶ A logical partition *must not be able to* access, copy, duplicate, read, or write data stored in a processor core allocated to another logical partition.
- ▶ A logical partition *must not be able to* deny, prevent, or influence another logical partition to get legitimate access to a processor core.
- ▶ A logical partition *must not be aware* of the existence of other processor cores allocated to other logical partitions.

4.2.3 Memory isolation

Memory areas allocated to different logical partitions must be totally isolated. In terms of memory isolation, the following objectives must be met:

- ▶ A logical partition *must not be able to* access, copy, duplicate, read, or write data stored in the memory area of another logical partition.
- ▶ A logical partition *must not be able to* deny, prevent, or influence another logical partition to get access to its legitimate amount of memory.
- ▶ A logical partition *must not be aware* of the existence of other memory areas allocated to other logical partitions.

4.2.4 I/O isolation

I/O devices allocated to different logical partitions must be isolated and operate independently. In terms of I/O isolation, the following objectives must be met:

- ▶ A logical partition *must not be able to* access, read, write, or alter performance of I/O devices allocated to another logical partition.
- ▶ A logical partition *must not be able to* deny, prevent, or influence another logical partition access to its legitimate I/O devices.
- ▶ A logical partition *must not even be aware* of the existence of I/O devices other than those explicitly allocated to it.

The following cases are typical examples of I/O isolation:

- ▶ An I/O device allocated to a logical partition must not be allowed to perform Direct Memory Access (DMA) operations in the address space of another logical partition.
- ▶ Failure of an I/O device allocated to a logical partition must not influence I/O activity on any other logical partition.

4.3 Domains of IBM Power processor cores

This section introduces concepts related to IBM Power processor cores and features that support logical partitioning on IBM Power Systems.

Processor cores have different *levels* of operation, sometimes referred to as *rings* or *domains*. Traditionally, processor cores have two domains: *application domain* and *kernel domain*. IBM Power processors introduced an extra domain named the *hypervisor*. This hypervisor domain plays a crucial role in virtualization and represents one of the core components of a complex virtualization framework that enables logical partitioning isolation, integrity, and security.

4.3.1 Application domain

The application domain is used by the applications running within logical partitions. Application code and data do not have access to any system resource. When applications need to get access to system resources, they make a request to an entity that is entitled to get access to those resources on their behalf. For example, in case of AIX operating systems, this entity is called a *kernel*.

This is the level with the lowest privileges. When running in this level, none of the privileges and resources associated with superior levels are accessible.

Applications running on the operating system make system calls to request services from the kernel. Access to system resources is allowed *only* by kernel services and applications rely on the kernel to get access to system resources.

4.3.2 Kernel domain

In existing traditional non-virtualized environments with two domains, the kernel domain will be the most privileged domain, and provide access to all resources. The kernel of the operating systems will execute in this domain. The kernel manages the entire operating system and maintains critical operating system resources such as virtual memory address space and memory page tables. The state of the operating system is referred to as *context* and the kernel is responsible for maintaining it. The kernel retains exclusive control over all system resources. The kernel has control over processor cores and has direct access to physical memory. The kernel controls, mediates, and ultimately is responsible for access to all I/O resources.

This domain is more privileged than the application domain. Applications running on the operating system are prevented from and cannot run in this more privileged state.

With the advent of virtualized environments, *multiple* operating systems have to share access to common resources such as processor cores, physical memory, and I/O devices. The kernel is still responsible for maintaining the context of operating system running in an individual logical partition and for managing some of the operating system critical resources.

However, the kernel of the operating system no longer has full control over processor cores, cannot access directly physical memory, and does not control and manage system I/O resources any more.

In virtualized environments the access to system-wide resources, such as processor cores, physical memory, and I/O devices is controlled and mediated by a new component called the *hypervisor*. Operating system kernels make *hypervisor calls* and request services from the hypervisor in the same way that a regular application running on the operating system makes a system call to request a service from the kernel.

The hypervisor retains the control over all critical system resources and operating system kernels have access to these resources *only* through hypervisor calls. The hypervisor is responsible for providing the kernel running in a logical partition access to partition resources and for ensuring the kernel access is limited to the resources the partition is entitled to.

4.3.3 Hypervisor domain

IBM Power processors provide an additional domain called the *hypervisor*. This is the most privileged level in which processor cores can run.

IBM Power Systems feature a *trusted firmware component* referred to as the POWER Hypervisor (see 4.5, “POWER Hypervisor” on page 65). It is the *only* software component that runs in hypervisor mode. Operating systems running in logical partitions are prevented from and cannot run in this privileged state.

IBM POWER processor cores have *special instruction sets* that can run only in hypervisor mode. These instruction sets can be used exclusively by the IBM POWER Hypervisor.

IBM Power processors have a *special set of hardware registers* that are accessible only in hypervisor mode. These registers are used to implement virtualization and maintain the isolation of logical partitions.

The POWER Hypervisor has full access to the entire system memory address space.

Transition to this privileged processor mode can be done only using specific POWER Hypervisor calls referred to as *hcalls*. Hypervisor calls are initiated in the kernel space of the operating systems running in logical partitions. The operating system kernel makes a hypervisor call just like a regular application running on the operating system makes a system call to request a service from the kernel.

For all hcalls, the POWER Hypervisor first validates that the partition that makes the hcall has been granted access to resources that are the subject of the hcall and then completes the request. This additional validation enforces the least privileged principle and ensures complete isolation of resources belonging to different partitions.

4.4 Processor core access modes

With IBM POWER technology, processor cores can access memory in several *modes*. For a more comprehensive description of various modes for accessing memory, see section 5.7 Storage Addressing of the *Power ISA Version 2.07* document, which is available at the following website:

<http://power.org>

For the sake of simplicity in this publication, these modes are divided into the following two classes:

- ▶ Real addressing mode
Real access mode implies that all addresses generated and used by the processor cores are used just as they are and no additional address translations are required.
- ▶ Indirect addressing mode
Indirect access mode implies that one or more levels of indirection are used and address translations are required.

4.5 POWER Hypervisor

Virtualization of IBM Power Systems relies on a firmware component referred to as the *POWER Hypervisor*. The POWER Hypervisor (PHYP) is the lowest layer of firmware that runs on IBM Power Systems and provides virtualization services to upper layers of firmware and software. The hypervisor enforces the underlying control mechanism that resides below the operating system layer and above the hardware layer. The layers above the POWER Hypervisor are specific to each supported operating system.

PHYP is responsible for initialization of all IBM Power Systems, even those that contain just a single partition. It is loaded by the flexible service processor (FSP) into the physical memory when the systems are powered on. PHYP is responsible for system initial program load (IPL), initialization of all hardware devices, and dispatching logical partitions. POWER Hypervisor is allowed to access physical memory in real addressing mode.

In addition to virtualizing and partitioning physical hardware, the POWER Hypervisor also provides support for pure virtual devices that have no underlying physical hardware such as virtual LAN adapters and virtual console devices.

Additionally, POWER Hypervisor provides other high level functions such as system error reporting, local partition management, and hardware error detection and recovery.

POWER Hypervisor uses a paravirtualization strategy. This means that operating systems are *enhanced* to become hypervisor-aware and use a well-defined set of interfaces.

Partition firmware and operating system functions such as interrupt management or virtual memory management use these hypervisor interfaces to get access to hardware resources. Hypervisor interfaces also provide access to pure virtual devices such as virtual LAN adapters and inter-partition communication channels.

The POWER Hypervisor maintains a copy of the context of each logical partition. When a logical partition is dispatched, the hypervisor deploys the context of the partition and then cedes the control of processors to the partition operating system. The POWER Hypervisor maintains the control over all critical resources for both dedicated and micropartitions. To achieve this, the hypervisor maintains system-wide special data structures such as the Translation Control Entry (TCE) table. These data structures are accessible only in hypervisor mode by using special instruction sets and special hardware registers that are accessible only in hypervisor mode.

The POWER Hypervisor also manages data structures such as Partition Page Tables (PPT) for each logical partition. It maintains these data structures on behalf of logical partitions using privileged instructions and registers available only in hypervisor mode.

All critical data structures are kept in memory areas that are accessible to *only* the Hypervisor.

For a better understanding of virtualization technology and PHYP, see the *Power ISA Version 2.07* document, which is available at the following website:

<http://power.org>

4.5.1 POWER Hypervisor integrity

POWER Hypervisor runs in the protected hypervisor mode on IBM Power processor cores. IBM is the sole developer and product owner of the POWER Hypervisor. This implies that other parties cannot install code or components into the hypervisor.

IBM Power Systems have a separate *service processor* called the flexible service processor (FSP). The FSP has its own separate processors, memory, buses, and I/O such as flash memory. FSP is not accessible from the logical partitions. The FSP is connected to the HMC through an encrypted communication channel.

When a new version of firmware is available, the image is retrieved through the HMC. The HMC sends the new version to the FSP for validation. The FSP validates the digital signature used to sign the new firmware version to ensure that it has not been tampered with and was

indeed originated by IBM. If the validation fails, the update does not occur. If the validation is successful, the flash memory on the FSP is updated with the new version.

Because the flash memory resides on the FSP, the LPARs and even the POWER Hypervisor cannot get access to, alter, or modify the code.

4.5.2 POWER Hypervisor and processor core sharing

When logical partitions are configured to use shared processor cores, the POWER Hypervisor maintains the core status for each logical partition by saving context in a facility called a *virtual processor*.

The POWER Hypervisor uses virtual processors to encapsulate all data relevant to logical partitions state from a processor core perspective at a specific point in time. Virtual processors reflect the status and content of processor core registers and are part of the context of a logical partition.

Virtual processors introduce an abstract layer between the operating system and the physical cores. The operating systems perceive only virtual processors and act like they have physical cores available. Processes run by the operating system are assigned to virtual processors, which are dispatched by the POWER Hypervisor on actual physical cores. A hypervisor call made by an operating system that is running in a logical partition is run *only* on the same processor core on which the operating system was dispatched when the call was initiated. All hypervisor calls are, by design, tagged with the ID of the logical partition initiating the call. This ensures that hcalls cannot access other data of other logical partitions.

Virtual processors represent a key component of IBM PowerVM that enables IBM Power Systems to support processor core sharing and micropartitioning. *Both* dedicated and micropartitions use virtual processors.

The POWER Hypervisor controls which logical partitions and which virtual processors are dispatched on processor cores. When a context switch occurs, the current status of a physical core is saved by the POWER Hypervisor into a virtual processor state and stored in a memory area accessible *only* by the hypervisor. The POWER Hypervisor ensures that all hardware registers and cache lines are cleared to prepare the loading of the next context. PHYP then selects the virtual processor that will be dispatched next, reads the relevant data from it, loads the processor core hardware registers with the new values and finally cedes the control of the physical core to the operating system. This mechanism allows multiple virtual processors from multiple logical partitions to share the same processor core and maintain logical partition isolation.

The newly dispatched partition is allowed to run on the physical processor core for a predetermined amount of time after which the POWER Hypervisor regains the control of the core. There might be cases when the operating system voluntarily relinquishes the control of the processor core through a system call to the POWER Hypervisor.

IBM Power processors have several mechanisms that ensure that the POWER Hypervisor ultimately gains the control of processor cores. These features provide logical partitions protection against potential denial of service attacks, crashes, and failures of other logical partitions.

For example, the hypervisor decremter is a hardware facility that provides the POWER Hypervisor with a timed interrupt, *independent* of the logical partition activity. This ensures that the POWER Hypervisor eventually regains the control of processor cores and prevents any denial of service from a potentially malicious logical partition.

4.5.3 POWER Hypervisor and memory sharing

Memory sharing between logical partitions is strictly prohibited. It is not possible for a logical partition to access, copy, duplicate, alter, read, or write data stored in the memory space of another partition. A logical partition is not even aware of the existence of other memory areas allocated to other logical partitions.

The POWER Hypervisor manages and controls physical memory. It divides the physical memory into Physical Memory Blocks (PMB). Some of the PMBs are used only for POWER Hypervisor data and logical partitions cannot get access to them. Other PMBs are allocated to logical partitions according to their entitlements. For *each* logical partition, the POWER Hypervisor allocates a Partition Page Table (PPT) at boot time. All PPTs are stored in a memory area that is *different* from all memory areas allocated to logical partitions so the PPTs are accessible only to the POWER Hypervisor. Operating systems cannot read and write the content of PPT entries. Operating systems can get access to and update the content of the PPTs *only* by making hcalls.

The POWER Hypervisor *mediates* operating systems access to physical memory and *transparently* translates memory addresses used by the operating systems into physical memory addresses. This translation is facilitated by special hardware registers. The hardware logic prevents the modification of these special registers and they are accessible only to the POWER Hypervisor.

POWER Hypervisor design ensures that physical memory ranges that are assigned to different logical partitions do not overlap. POWER Hypervisor prevents each operating system running in a logical partition from accessing any memory area that was not allocated to it. Logical partitions rely on the POWER Hypervisor as a trusted entity to facilitate, mediate, and control access to memory.

The mechanisms used by the POWER Hypervisor to manage and control the access to shared memory are at the core of other IBM PowerVM features such as Active Memory Sharing or Active Memory Expansion.

4.5.4 POWER Hypervisor and I/O sharing

IBM Power Systems use network and disk adapters inserted in PCI slots located in I/O drawers. Communication with I/O devices uses memory buffers and address translation mechanisms similar to those used for memory sharing.

Again, it is the POWER Hypervisor that translates the addresses used by device drivers into physical memory addresses. POWER Hypervisor ensures that physical memory ranges assigned to and used by different device drivers do not overlap. Additionally the POWER Hypervisor enforces a strict control over partition ownership of I/O devices.

PCI devices use Direct Memory Access (DMA) operations to transfer data between adapters and memory. PCI device drivers read and write directly to the I/O space associated to PCI devices. Every PCI card adapter has its own separate PCI busses. PCI host bridges translate the addresses generated by I/O devices into physical memory addresses using a Translation Control Entry (TCE) table. During an IPL, the POWER Hypervisor allocates memory in the TCE table for each PCI slot. The size of the TCE table depends on the number of I/O drawers.

TCE table is owned and managed by the POWER Hypervisor. Similar to PPTs, TCE table is maintained and protected in a hypervisor owned region of memory inaccessible to logical partitions. The address translation mechanism using TCE table is similar to virtual memory addressing using PPTs. Similar to PPTs, TCE table can be updated by using *only* hypervisor calls.

In contrast with PPTs being associated to individual partitions, TCE is *unique* to the IBM POWER system and references all I/O slots. The association of an individual physical device with a partition is done indirectly by identifying the partition that owns the slot. The PCI device slot must be present in the logical partition profile before the POWER Hypervisor allows the partition to create or update an entry in the TCE table. In this manner the least privileged security principle is enforced.

When logical partitions share I/O resources, a separate service logical partition called a *Virtual I/O Server* is created. This special type of logical partition acts as an appliance that owns the shared I/O resources such as slots, adapters, and devices. The Virtual I/O Server (VIOS) logical partition is subject to the same restrictions as all other logical partitions. The data is transferred between regular logical partitions and VIOS partition by the POWER Hypervisor. For more information about the VIOS, see 4.9, “Virtualization of I/O devices” on page 76.

4.6 Memory isolation

This section shows you how memory isolation between multiple logical partitions is enforced. Because the word memory is used in so many contexts, definitions of the following terms that relate to memory are provided:

- ▶ 4.6.1, “Effective memory” on page 70
- ▶ 4.6.2, “Virtual memory” on page 70
- ▶ 4.6.3, “Physical memory” on page 71
- ▶ 4.6.4, “Real memory” on page 71
- ▶ 4.6.5, “Logical memory” on page 72
- ▶ 4.6.6, “Partition page tables” on page 72

Figure 4-1 on page 70 illustrates the relationships between effective, virtual, and physical memory spaces.

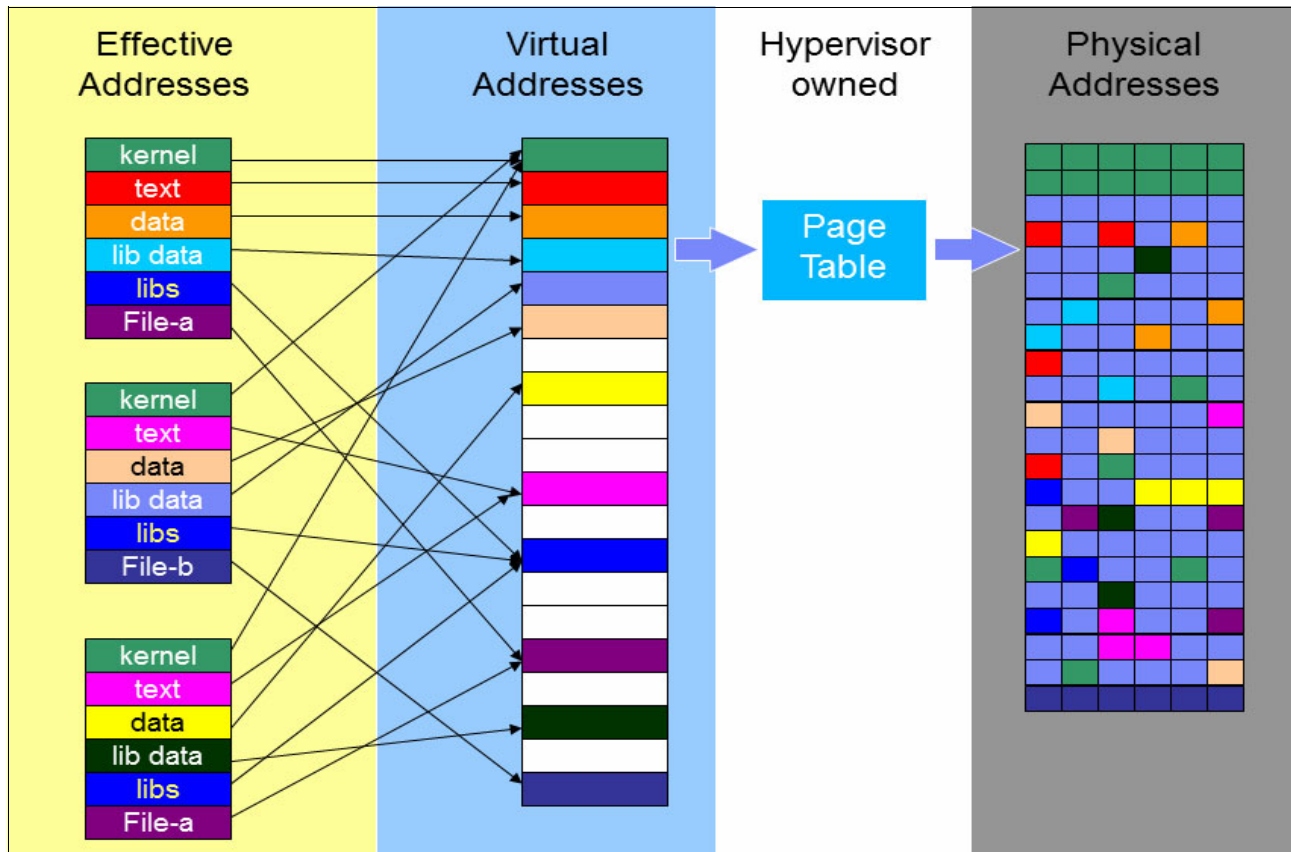


Figure 4-1 Relationship between effective, virtual, and physical memory spaces

4.6.1 Effective memory

The term *effective memory* refers to the address space available to individual processes. Each process has its own unique *effective address space*. This allows every process to run as though it is the only process running on the system and has enough memory resources available. Processes are unaware about the hardware details of the systems on which they run. Memory addresses used by applications are referred to as *effective addresses*.

The effective address space is divided into a predefined number of regions called *segments*. Each segment contains a predefined several smaller size regions called *pages*. The layout of the effective address space depends on the programming model. For example, 32-bit AIX processes can have 16 segments. The 64-bit AIX processes have an effective address space that can include up to 2^{36} segments.

4.6.2 Virtual memory

The term *virtual memory* refers to the memory space provided by and managed by the operating system. The *virtual address space* allows the operating system to address a memory space larger than the amount of available physically memory in the system, and provides each application an effective address space.

Each instance of an operating system has its *own* virtual address space. Similar to the effective address space, the virtual address space is divided into segments, which in turn are divided into pages.

Effective addresses used in each process effective address space are mapped by the operating system to virtual addresses in the operating system virtual address space. Segments from the effective address space are mapped to segments in the virtual address space. Applications are *not aware* of the fact that addresses used by them are translated into virtual addresses. Applications access their data using indirect addressing mode.

4.6.3 Physical memory

The term physical memory refers to the tangible DIMMs that are installed in IBM Power Systems. The amount of available RAM constitutes the *physical address space*. Physical memory is divided into pages.

Physical memory provides support for virtual memory because ultimately all virtual memory pages must be mapped to physical memory pages. Because the whole virtual address space cannot be accommodated by the existing physical address space, the operating system also uses disks to store data.

The operating system writes the least used application data from physical memory to disk and reads back the data from the disk in memory when data is needed. This mechanism allows the operating system to behave as though it has more physical memory available than it really has. The component of the operating system that performs these operations is called the Virtual Memory Manager (VMM). The VMM uses translation *page tables* to do these conversions. Page tables always reside in physical memory and are *unique* to each operating system instance.

The physical address space also accommodates memory areas used by hardware components such as network or disk adapters. Depending on the hardware implementation and restrictions, memory address ranges assigned to hardware adapters can be dispersed throughout the physical address space, which might result in a fragmented physical memory address space. For example, if a PCI adapter device requires DMA operations, the DMA address of the device is mapped to a specific physical memory address range by the PCI host bridge. Most VMMs of modern operating systems are designed to handle non-contiguous physical memory address ranges.

4.6.4 Real memory

Traditionally, operating systems such as AIX or Linux would require certain amounts of contiguous memory not to be translated, typically for bootstrapping, kernel internal data structures, or interrupt vectors. Addresses used by the processor cores for instructions and data in these areas would not be translated, which means the processor cores would access these memory areas in real addressing mode.

The term *real memory* refers to the memory that can be accessed when running in real addressing mode. The *real memory address space* must start with address 0.

In traditional non-partitioned environments, the entire amount of physical memory is assigned to a single partition so all physical memory can be addressed directly in real addressing mode. In such cases, a one-to-one relationship exists between physical memory space and real memory space.

However, in a partitioned environment, multiple partitions share physical memory. Each partition must have its own memory areas, accessible in real addressing mode, and its own real memory space that starts with address 0 (zero).

4.6.5 Logical memory

Multiple partitions can be provided distinct address spaces accessible in real addressing mode and starting at address 0 by introducing the concept of *logical memory*.

Logical memory is an abstract representation that provides a contiguous memory address space to each logical partition. Multiple physical memory blocks are grouped together and presented to logical partitions as a contiguous memory address space. The operating system perceives the logical memory as though it was real memory. The logical memory address space starts with address 0.

When a partition is started, the POWER Hypervisor assigns the logical partition a unique real mode *offset* and a *range*. These values map partition logical memory to a physical memory address range that is exclusively assigned to the logical partition. The value of the range is determined by the values specified in partition profile. These values are set in the hardware registers controlled by the POWER Hypervisor on each processor core used by the partition. By using these values, the hardware *transparently* redirects any partition logical memory address to the correct location in the physical memory by adding the offset to the address. Partition processors can then access the memory data in real addressing mode. Because hardware registers that maintain these values are accessible only to the POWER Hypervisor, they cannot be accessed or modified by the operating system running in the logical partitions.

Any attempt to access a real memory address outside the assigned range results in an addressing exception interrupt that is treated by the logical partition operating system exception handler.

4.6.6 Partition page tables

In traditional non-partitioned environments, the responsibility to manage the page tables goes to the VMM. The VMM accesses page tables in real addressing mode and the operating system maintains a processor register that points to these tables.

However, in partitioned environments, there are *different* sets of page tables for each logical partition. Any logical partitions has access *only* to its own page tables. This is the reason why partition page tables are kept in a hypervisor-owned region of memory that is inaccessible to logical partitions. The processor registers that control the access to page tables can be accessed only in hypervisor mode.

When an operating system running in a partition needs to access partition page tables, it makes a hypervisor service call and switches to hypervisor mode. The POWER Hypervisor is what accesses the page tables on behalf of the partition and retrieves or updates the data.

Entries in the page tables map logical address regions to physical address regions. The POWER Hypervisor ensures that partitions do not share any region of physical memory. Operating systems running in logical partitions cannot determine where their memory has been physically allocated.

Figure 4-2 illustrates how the POWER Hypervisor manages and isolates memory areas assigned to different logical partitions.

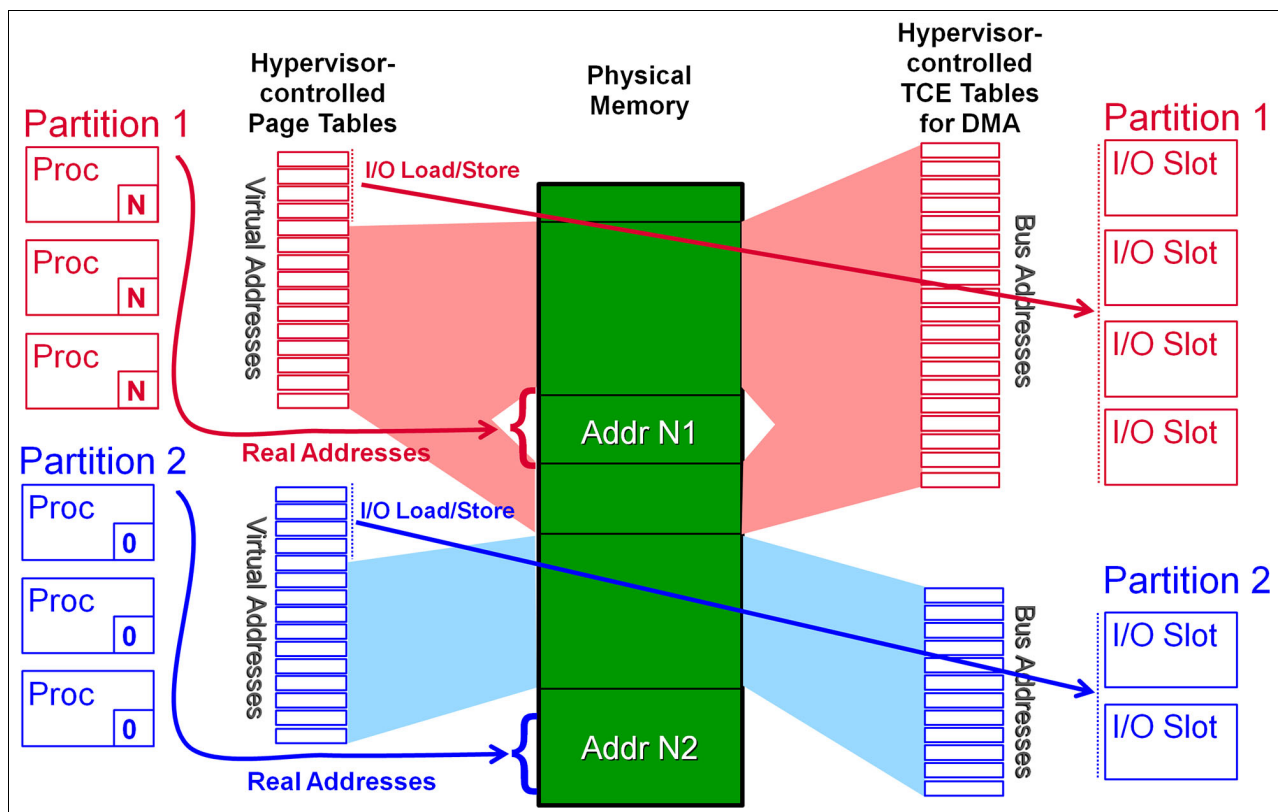


Figure 4-2 Isolated memory areas assigned to partitions managed by the hypervisor

4.7 I/O isolation

This section explains how I/O isolation between multiple logical partitions is enforced by the POWER Hypervisor.

At platform IPL time, the POWER Hypervisor allocates memory in the TCE table for each PCI slot. DMA ranges are assigned to the every PCI Host Bridge (PHB) by the POWER Hypervisor and communicated to the operating systems running in logical partitions through a configuration structure passed to the operating system at boot time. The DMA ranges for every PHB are maintained in special hardware registers by the POWER Hypervisor.

Logical partitions can perform various operations related to specific PCI adapters. They use hypervisor service calls to create, delete, and update TCE entries. POWER Hypervisor verifies first if the partition is authorized to access TCE entries associated to the specific PCI adapters.

TCEs are used to perform the translation of PCI bus addresses to a valid physical memory address. The TCE table is located in a POWER Hypervisor owned memory area that is accessible only to the POWER Hypervisor. A TCE includes the following data:

- ▶ Platform Memory Page Address
- ▶ Authority/Protection bits

This data is used in conjunction with the PCI bus address. The PCI bus address includes the following data:

- ▶ TCE Table offset
- ▶ Memory Page offset

The PCI bus address is sent to the I/O hardware to perform DMA operations. When the PHB performs the address translation using the PCI bus address, the following steps occur:

1. The TCE Table Offset is added to the TCE Base Address Register, which is maintained and updated by the POWER Hypervisor, and points to the beginning of the TCE table for that PHB.
2. The TCE is fetched from memory.
3. The PHB combines the memory page address from the TCE with the memory page offset from the PCI bus address to create the full physical memory address.
4. The PHB then validates the protection bits are set correctly for the requested DMA operation type.

During runtime, the PHB validates DMA address usage using the values kept in the hardware registers. Operating systems running in logical partitions prepare for DMA operations by invoking a hypervisor interface to map a TCE corresponding to a specific I/O adapter. The operating system specifies a TCE offset to be mapped for I/O to a memory page within the logical partition. The hypervisor then ensures that devices cannot be used to access data from other logical partitions and provides a secure DMA by validating the following information:

- ▶ The logical partition owns the I/O adapter associated with the TCE.
- ▶ The memory page to be mapped to the TCE is a valid memory page for the requesting logical partition.

After the I/O adapter ownership and memory page validity are verified, the hypervisor maps the requested TCE to the memory page with the protection specified by the operating system.

DMA operations are performed by the PHB within the I/O subsystem. The PHB translates a PCI bus address into a memory address through the use of a TCE.

TCEs are owned by the hypervisor and the system address given by an operating system are validated *prior* to being used by the I/O hardware for a DMA operation. Optimal performance is achieved by the I/O hardware performing DMA reads and writes by directly accessing logical partition memory.

Figure 4-3 on page 75 illustrates how memory areas used for I/O operations are managed by the hypervisor.

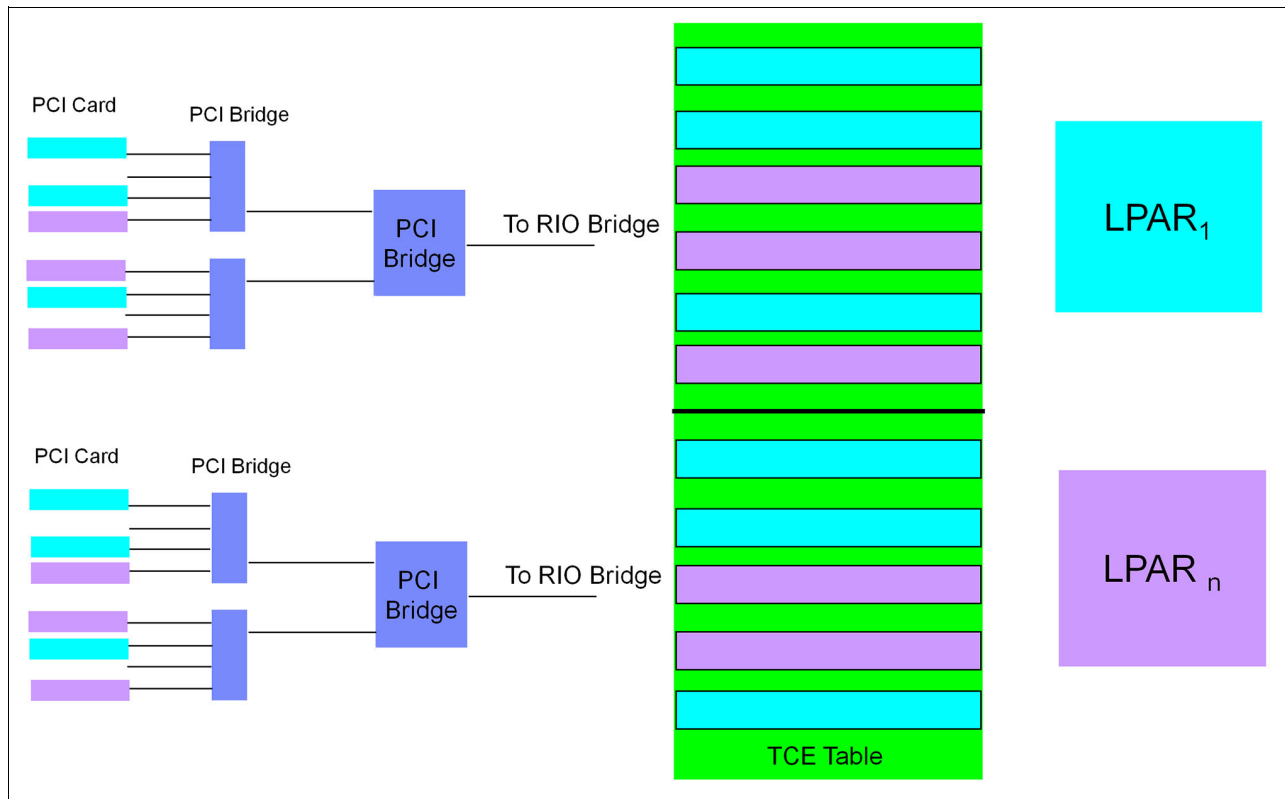


Figure 4-3 TCE table and logical partitions I/O memory areas

4.8 Logical partitions (LPARs)

In cloud and multitenant environments based on IBM Power Systems, securing the configuration and management of logical partitions is important. POWER Hypervisor and IBM Power Systems are designed to ensure the least privileged security is enforced at all times. Logical partitions are allowed access only to resources that have been explicitly assigned to them. Resources that are not explicitly assigned cannot be accessed. Logical partitions are unaware of the existence of other logical partitions on the same physical system.

4.8.1 LPAR management

Logical partitions profiles are configured and managed using the IBM Hardware Management Console (HMC). The HMC connects to the FSP and stores partition configuration data in the NVRAM area of the FSP. Configuration data is used by the POWER Hypervisor to determine which resources are assigned to each logical partition.

Only the HMC and the POWER Hypervisor can get access to data the stored in the NVRAM. Logical partitions do not have access to the FSP, therefore they cannot modify their profiles to gain unauthorized access to additional resources. Moreover, any logical partition cannot alter profiles of other partitions and prevent them from using resources they are entitled to.

Although the HMC is used to service various hardware components of IBM Power Systems, it has no access to memory and I/O devices assigned to logical partitions. Access to memory and I/O devices is carefully controlled and mediated by the POWER Hypervisor.

For more information about the HMC, see Chapter 3, “IBM Hardware Management Console (HMC) security” on page 39.

4.8.2 LPAR operating systems

The POWER Hypervisor controls LPAR access to memory and translates virtual memory addresses into physical memory addresses. This mechanism confines the accessible memory areas and prevents the operating systems from accessing any unauthorized memory area.

The POWER Hypervisor also ensures that all hypervisor calls invoked by the operating system kernel refer only to resources that are assigned to the logical partition.

There are also special hypervisor interfaces and service calls that allow operating systems to open virtual TTY consoles and collect debug or performance data.

4.9 Virtualization of I/O devices

IBM PowerVM provides support for virtualization of I/O devices on IBM Power Systems. Virtualization and sharing of I/O devices are facilitated by a virtual appliance referred to as the *Virtual I/O Server (VIOS)*.

VIOS is an AIX based software appliance that runs in its own logical partition that is subject to all security constraints that apply to all other logical partitions. The VIOS takes advantage of the underlying AIX hardware device drivers to access physical devices. Multiple instances of VIOS can run simultaneously on IBM Power Systems.

Similar to the HMC, the VIOS has a restrictive command-line interface that is used for management and administrative operations and is not intended to run any user application.

The VIOS is managed by a set of command-line utilities accessible to a user named *padmin*. The utilities are implemented as binaries and shell scripts. The *padmin* user runs under a restricted shell. There may be times, such as configuring storage access, when the VIOS operation requires running specific commands with root privileges. This is achieved by invoking the commands under a *setuid* program named *ioscli*. This program accepts only a restricted set of specific commands. Both *ioscli* and restricted shell are used to enforce the restrictions on the VIOS.

Logical partitions and VIOS partitions use virtual adapters to establish inter-partition communication under the control of the POWER Hypervisor. VIOS has the ownership of physical I/O resources, provides virtualization of I/O devices, and ensures the separation of data flows for individual logical partitions.

Access to disks is provided by implementations of Virtual SCSI and Virtual Fibre Channel. Both implementations use a client/server model in which virtual server adapters are owned by the VIOS partition and virtual client adapters are owned by the client partition. Virtual server adapters control the access to real physical devices.

System administrators must configure the VIOS and explicitly dedicate physical devices to the logical partition hosting the VIOS. Because the VIOS is confined to logical partition limits, it cannot access physical or logical resources assigned to other logical partitions. Conversely, other logical partitions do not have access to physical or virtual resources owned by the VIOS.

4.9.1 Disk access for logical partitions

Both Virtual SCSI and Virtual Fibre Channel implementations use a client/server model in which VIOS server adapters control and mediate access to real physical devices. VIOS, virtual device drivers, and POWER Hypervisor work together to ensure that each logical partition has access only to its own data.

VIOS controls the data flow and performs DMA operations to transfer data between logical memory ranges assigned to different partition; the POWER Hypervisor controls the mapping of logical to physical memory ranges.

4.9.2 Network access for logical partitions

The POWER Hypervisor provides connectivity between different logical partitions using Virtual Ethernet adapters.

The POWER Hypervisor provides a software implementation of an Ethernet switch that is compliant with standard IEEE 802.1Q. This switch allows operating systems running in logical partitions to communicate using standard networking protocols. IBM Power Systems can be configured to use multiple virtual switches.

Logical partitions can have multiple Virtual Ethernet adapters. Each Virtual Ethernet adapter is connected to a POWER Hypervisor software switch. The POWER Hypervisor is invoked for transmission of each Ethernet frame and copies the data between logical partition memory areas. Virtual Ethernet adapters and POWER Hypervisor switch provide the means for efficient inter-partition communication at memory speed. Because the virtual switch functions are provided by the POWER Hypervisor, communication between logical partitions does not require configuration of a VIOS.

Each adapter is assigned a unique MAC address derived from the system serial number, LPAR ID, and adapter ID. Similar to physical switches, the virtual switch allows for creation of multiple VLANs to separate network traffic between logical partitions. The IEEE 802.1Q standard enforces that a VLAN ID is inserted into each Ethernet frame. The software switch restricts the Ethernet frames to switch ports authorized to receive frames with specific VLAN IDs. Ports of the virtual Ethernet switch can be configured to belong to multiple VLANs. For Virtual Ethernet adapters, the same security considerations apply as for physical network adapters. VLAN support is one of the mechanisms used to segregate traffic between logical partitions.

VIOS Shared Ethernet Adapter (SEA) function provides connectivity to external networks for logical partitions that have only Virtual Ethernet adapters. The SEA acts like a layer 2 bridge to the physical adapters.

The VIOS was designed to limit the exposure of I/O devices to the POWER Hypervisor. This provides the POWER Hypervisor simplicity and security.

Restricting VIOS access: A preferred practice is to restrict VIOS access to a separate administrative network, similar to HMC and FSP. If using a dedicated physical Ethernet adapter is not possible, the IP address of the VIOS server should be on a separate VLAN.

IBM PowerVM can provide support for virtualization of I/O devices on IBM Power Systems using single root I/O virtualization (SR-IOV). SR-IOV is an extension to the PCI Express (PCIe) specification that allows multiple instances of operating systems to simultaneously share access to a PCIe adapter with minimum or no involvement from the Power Hypervisor. From the logical partition perspective, the adapter appears to be a physical I/O device.

IBM PowerVM provides support for SR-IOV capable I/O devices on selected IBM Power Systems. IBM PowerVM provides a feature-named logical port that defines the characteristics and capabilities for a fraction of a physical port of an SR-IOV capable I/O device. A logical port for a logical partition is created using the HMC and is assigned a specific capacity that determines the desired percentage of the physical port bandwidth, which the logical partition is entitled to use. Power Hypervisor uses the configuration of the logical port to manage platform firmware resources and to configure the I/O device.

Multiple logical partitions access their corresponding share of the adapter by using their own VF device drivers. From the logical partition perspective, the VF is considered a single-function, single-port adapter and is treated like physical I/O.

Virtual network interface controller (vNIC) is a new virtual adapter type that leverages SR-IOV and enables logical partitions network connectivity.

For more information about SR-IOV, see *IBM Power Systems SR-IOV: Technical Overview and Introduction*, REDP-5065.

IBM POWER7+™ and POWER8 systems provide isolation domains for I/O devices attached through a PCI Express (PCIe) I/O bus. If an I/O device, attached through the PCIe I/O bus, is assigned to an individual logical partition, then the isolation domain is the I/O device and partition.

If the I/O device is enabled for sharing by multiple logical partitions using SR-IOV technology, then there is a hierarchy of isolation domains and the lowest level of isolation is a virtual function (VF). When an SR-IOV logical port is activated, either through logical partition activation or through a DLPAR operation, a VF is associated with the logical port to allow the logical partition to access the PCI device.

The VF is the lowest level entity that can be assigned to a logical partition and it cannot be shared with other logical partitions. For an SR-IOV, the isolation domain is the VF and the logical partition to which it is assigned. The isolation is accomplished using hardware features that are managed by Power Hypervisor. One of these features is a TCE table. Each VF has its own TCE table.

TCEs are used to perform the translation of PCI bus addresses to a valid physical memory address. The TCE table is located in a POWER Hypervisor owned memory area that is accessible only to the POWER Hypervisor. The TCE table controls how the VF gets access to system memory.

Power Hypervisor validates that the memory addresses in the TCE are mapped to the logical partition to which the VF is assigned. A VF can access system memory only through the TCE table. Therefore, a VF can access only the memory of the logical partition to which the VF is assigned.

4.10 Security of DLPAR operations

Dynamic LPAR (DLPAR) allows IBM Power systems to dynamically add and delete selected system resources from logical partitions while they are running. Resources subject to DLPAR operations include processor cores, memory, physical I/O adapters, and virtual adapters such as virtual SCSI adapters or virtual Fibre Channel adapters. Adding and removing resources from logical partitions can be performed *without* restarting the logical partitions or the operating systems running in logical partitions.

This continuous availability is the direct result of the following features available with IBM Power systems:

- ▶ HMC

The HMC provides the interface used for adding and removing resources.

- ▶ Operating systems

The operating systems supported by IBM Power systems are enhanced to be capable of dynamically acquiring and releasing resources.

- ▶ POWER Hypervisor

The POWER Hypervisor manages the assignment of resources to logical partitions and mediates logical partitions access to the resources.

For a comprehensive description of DLPAR capabilities of IBM Power systems see the following Redbooks publications:

- ▶ *IBM PowerVM Virtualization Introduction and Configuration*, SG24-7940
- ▶ *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590

The security of DLPAR operations is based on the tight integration of the following facts:

- ▶ Operating systems running in logical partitions have been enhanced and improved to become *hypervisor-aware*.
- ▶ POWER Hypervisor *functionalities* allow for enforcing the control over system resources at all times.
- ▶ POWER Hypervisor *design* is flexible and allows for dynamic changes of access control over system resources.

To help you better understand how DLPAR operations work, this section depicts how a common DLPAR operation such as reallocation memory from a running AIX logical partition to another running AIX logical partition takes place. This process consists of the following steps:

1. The HMC user interface is used to select the amount of memory that will be removed and the logical partition that releases the memory. The HMC validates the required operation against the logical partition profile. Removing memory from a logical partition is not allowed if the remaining amount of memory is lower than the minimum amount specified in the active profile.
2. The HMC explicitly requests the operating system to release the specified amount of memory.
3. The operating system explicitly releases the memory and updates the operating system context accordingly. In the case of the AIX operating system this operation is performed by the VMM.
4. The POWER Hypervisor takes control of the released memory and updates all hypervisor-owned data structures that reflect the logical partition new context.

5. The POWER Hypervisor temporarily prevents all other logical partitions from getting access to the resource that has been released.
6. The POWER Hypervisor ensures the resource that has been released can be safely allocated to the receiving logical partition. The memory is scrubbed upon release. I/O physical adapters undergo a hardware reset.
7. The HMC validates the memory addition against the profile of receiving logical partition. Adding memory to a logical partition is not allowed if the new total amount of memory is bigger than the maximum amount of memory specified in the active profile.
8. The memory is allocated to the receiving logical partition by the POWER Hypervisor. All hypervisor-owned data structures are updated to reflect the logical partition new context.
9. The HMC explicitly notifies the operating system running in the receiving logical partition about the availability of additional memory and requests the operating system to acquire it.
10. The operating system explicitly acquires the memory and updates the operating system context accordingly. In the case of AIX operating system, this operation is again performed by the VMM.

4.11 IBM PowerVM security management with PowerSC

This section introduces the IBM PowerSC™, a trusted solution that provides useful features to significantly simplify security and compliance in complex cloud and multitenant environments that are based on IBM Power Systems.

IBM PowerSC provides the following functionalities:

► Security and compliance automation

The security and compliance automation feature or IBM PowerSC automates configuring, monitoring, and auditing of security and compliance. It provides predefined security profiles that incorporate security settings tailored for the following standards:

- Payment Card Industry Data Security Standard (PCI DSS)
- Sarbanes-Oxley Act and COBIT compliance (SOX/COBIT)
- US Department of Defense (DOD) Security Technical Implementation Guide (STIG)
- Health Insurance Portability and Accountability Act (HIPAA)
- North American Electric Reliability Corporation (NERC)

The profiles can be deployed with default values or can be further customized to meet particular requirements for individual customer environments.

► Trusted boot

The trusted boot feature assesses the boot image, operating system, and applications, and attests their trust using the virtual Trusted Platform Module (vTPM) technology. It provides the means for the administrator to ensure that systems boot and run only trusted code.

► Trusted firewall

The trusted firewall feature ensures proper network isolation between multiple logical partitions by implementing a firewall in the hypervisor. It enables direct routing between specified VLANs that are controlled by the same VIOS and saves the bandwidth of physical network adapters.

- ▶ Trusted logging

The trusted logging feature allows for consolidation of AIX operating system logs on a VIOS in real time. It provides tamper-proof logging and a convenient solution for log backup and management. Operating system administrators cannot alter the logs, logs are persisted even if the logical partition no longer exists. This feature can be used when you plan for system auditing and compliance.

- ▶ Real-time compliance

The real-time compliance features allows for definition and enforcement of security policies. It monitors and enables an AIX logical partition to maintain security. Moreover, this feature can provide alerts when a change to the system violates any rule that is part of the configuration policy.

- ▶ Trusted Network Connect and Patch management

The trusted network connect and patch management feature is useful in large cloud and multitenant environments where a large number of logical partitions must be regularly updated and maintained. It verifies whether AIX logical partitions are at a specified software and patch level. This feature can also provide alerts when logical partitions that are not up-to-date are created, or when a security patch that might affect multiple logical partitions is issued.

- ▶ Trusted surveyor

The trusted surveyor feature monitors compliance with network isolation requirements. It allows for definition and enforcement of network segregation policies. The trusted surveyor indicates when VLAN configurations are not in the correct state.

For more information about IBM PowerSC, see the IBM PowerSC website:

<http://www.ibm.com/systems/power/software/security/>

4.12 Secure Logical Partition Mobility

This section introduces the IBM PowerVM features that can be used to securely move logical partitions between IBM Power Systems which is a common activity in cloud and multitenant environments. Relocation of logical partitions must be done in a secure manner, with maximum flexibility and agility while meeting stringent service-level agreements (SLAs).

4.12.1 Live Partition Mobility

Live Partition Mobility (LPM) is a PowerVM feature that you can use to migrate logical partitions, which are running IBM AIX, IBM i, and Linux operating systems, from one physical server to another without disrupting the applications that run on the respective operating systems.

The migration operation maintains complete system transactional integrity and transfers the entire environment, including processor state, memory content, attached virtual devices, and connected users.

For a comprehensive description of LPM, see the following Redbooks publications:

- ▶ *IBM PowerVM Virtualization Introduction and Configuration*, SG24-7940
- ▶ *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590
- ▶ *IBM PowerVM Enhancements What is New in 2013*, SG24-8198

Note: The Redbooks publication, *IBM PowerVM Live Partition Mobility (Obsolete - See Abstract for Information)*, SG24-7460, is no longer valid and the contents of this publication should no longer be considered accurate. Information from this publication has been updated and migrated to the publications in the previous list.

When using LPM, the data transferred over the network is not protected, by default. However, administrators can perform secure LPM operations during which data is transferred through an Internet Protocol Security (IPSec) tunnel created between Virtual I/O Servers located on the source and target physical systems.

IPSec protocols secure network connections between IP addresses and port pairs. IPSec provides data origin authentication, data integrity, data confidentiality, key management, and management of security associations.

4.12.2 Practical scenario for secure LPM

This section provides a practical example of using secure LPM for a logical partition between two physical systems that are managed by the same HMC.

To secure LPM PowerSC security, profiles are applied on the Virtual I/O Servers. Applying the profiles provided by PowerSC mandates the creation of IPSec tunnels. This scenario shows how to secure LPM a logical partition between two IBM Power Systems.

This example uses the following environment:

- ▶ Source IBM Power Systems: model 8233-E8B and serial number 061AB2P
- ▶ Target IBM Power Systems: model 8233-E8B and serial number 10DD51P
- ▶ HMC
- ▶ Source system uses VIOS, named p750_2_vio1, as a mover partition
- ▶ Target system uses VIOS, named p750_0_vio2, as a mover partition
- ▶ Logical partition named LPAR_2

Prepare the Virtual I/O Servers on the source and target systems by performing the steps shown in Example 4-1 on page 83. Do the same sequence of steps on the both source and target mover partitions, as follows:

1. Verify the software level on the first VIOS called p750_2_vio1 using the **ioslevel** command. List the model, serial number, and firmware level of the first VIOS.
2. Verify that PowerSC file sets are installed on the system using the **ls1pp** command.
3. List available security profiles that are shown as an XML file in the `/etc/security/aixpert/custom` directory. For this example, the DoD profile is used.
4. Enable the DoD profile by using the **pscexpert** command. The output of the command shows that 38 rules were processed and the new security level of the VIOS is DoD.
5. Verify whether the *ike* daemon that provides IPSec services is running by using the **lssrc -g ike** command.
6. Verify the existence of active IPSec tunnels by using the **ike cmd=list** command. Note that there are no active IPSec tunnels.

Example 4-1 Preparing the environment for secure LPM

```
$ uname -n
p750_2_viol
$ ioslevel
2.2.3.3
$ oem_setup_env
# prtconf|egrep "Model|Serial|Firmware"
System Model: IBM,8233-E8B
Machine Serial Number: 061AB2P
Platform Firmware level: AL730_127
Firmware Version: IBM,AL730_127
  Model Architecture: chrp
  Model Implementation: Multiple Processor, PCI bus
* vsa0                U8233.E8B.061AB2P-V1-C0                LPAR Virtual
Serial Adapter

# ls|pp -l|grep -i powersc
powerscExp.ice.cmds      1.1.3.0  COMMITTED  ICE Express Security Extension
powerscExp.license       1.1.3.0  COMMITTED  PowerSC Express Edition
powerscExp.ice.cmds      1.1.3.0  COMMITTED  ICE Express Security Extension
# cd /etc/security/aixpert/custom
# ls -l
total 488
-r-x-----  1 root    system    55601 May 17 06:27 DataBase.xml
-r-x-----  1 root    system    23563 May 17 06:27 DoD.xml
-r-x-----  1 root    system    21810 May 17 06:27 DoD_to_AIXDefault.xml
-r-x-----  1 root    system    21427 May 17 06:27 Hipaa.xml
drwx-----  2 root    system     256 May 17 06:27 Ja_JP
-r-x-----  1 root    system    56983 May 17 06:27 PCI.xml
-r-x-----  1 root    system    46391 May 17 06:27 PCI_to_AIXDefault.xml
-r-x-----  1 root    system    11369 May 17 06:27 SOX-COBIT.xml
drwx-----  2 root    system     256 May 17 06:27 ca_ES
drwx-----  2 root    system     256 May 17 06:27 cs_CZ
drwx-----  2 root    system     256 May 17 06:27 de_DE
drwx-----  2 root    system     256 May 17 06:27 es_ES
drwx-----  2 root    system     256 May 17 06:27 fr_FR
drwx-----  2 root    system     256 May 17 06:27 hu_HU
drwx-----  2 root    system     256 May 17 06:27 it_IT
drwx-----  2 root    system     256 May 17 06:27 ko_KR
drwx-----  2 root    system     256 May 17 06:27 pl_PL
drwx-----  2 root    system     256 May 17 06:27 pt_BR
drwx-----  2 root    system     256 May 17 06:27 ru_RU
drwx-----  2 root    system     256 May 17 06:27 sk_SK
drwx-----  2 root    system     256 May 17 06:27 zh_CN
drwx-----  2 root    system     256 May 17 06:27 zh_TW

# pscxpert -f DoD.xml
Processedrules=38      Passedrules=38  Failedrules=0   Level=DoD
      Input file=DoD.xml

# lssrc -g ike
Subsystem      Group      PID      Status
tmd            ike        5308620   active
cpsd           ike        13828338  active
iked           ike        5701750   active

# ike cmd=list
No tunnels match your request.
```

After the environment is prepared, perform the secure LPM of the logical partition LPAR_2 by following the steps shown in Example 4-2:

1. Use the **lssyscfg** command to identify the source and target IBM Power Systems managed by the HMC.
2. Use **lssyscfg** command to determine where logical partition LPAR_2 resides. The partition is currently running on the system p750_2.
3. Use the **migr1par** command to initiate the LPM of logical partition LPAR_2 from source system p750_2 to target system p750_0. Note that the LPM is initiated at 11:33.
4. Use the **lslparmigr** command to monitor LPM status. Note that at 11:34, the migration is in progress.
5. Use the **ike cmd=list** command to verify that on the p750_2_vio1 VIO, an IPSec tunnel was automatically created and activated.
6. Use the **ike cmd=list** command to verify that on the p750_0_vio2 VIO, the other end of the secure tunnel was automatically created and activated. This is confirmed by local ID and remote ID shown at both tunnel ends.
7. Use the **lslparmigr** command to monitor LPM status. Note At 11:42 the migration is completed.
8. Use the **ike cmd=list** command to verify the tunnel was deactivated and deleted on both Virtual I/O Servers.
9. Use the **lssyscfg** command to confirm the logical partition LPAR_2 is now running on the target system.

Example 4-2 Performing secure LPM

```
hscroot@hmc7:~>lssyscfg -r sys -F name
p750_2
p750_0

hscroot@hmc7:~>lssyscfg -r lpar -m p750_0 -F name
p750_0_powervc
p750_0_vio2
p750_0_vio1
hscroot@hmc7:~>lssyscfg -r lpar -m p750_2 -F name
LPAR_4
p750_2_vio2
p750_2_vio1
LPAR_2

hscroot@hmc7:~>date;migr1par -o m -m p750_2 -t p750_0 -p LPAR_2 &
Wed Sep 17 11:33:08 EDT 2014
[1] 29482

hscroot@hmc7:~>date;lslparmigr -r lpar -m p750_2|grep LPAR_2
Wed Sep 17 11:34:21 EDT 2014
name=LPAR_2,lpar_id=4,migration_state=Migration
Starting,migration_type=active,dest_sys_name=p750_0,dest_lpar_id=4,source_msp_name=p750_2_vio1,source_msp_id=1,dest_msp_name=p750_0_vio2,dest_msp_id=2,remote_manager=unavailable,remote_user=unavailable,bytes_transmitted=7253576759,bytes_remaining=1885446144

# uname -n;date;ike cmd=list
p750_2_vio1
Wed Sep 17 11:36:06 EDT 2014
```

Phase	Tun Id	Status	Local Id	Remote Id
1	1	Active	172.16.22.10	172.16.22.5
2	1	Active	172.16.22.10-172.16.22.10	172.16.22.5-172.16.22.5

```
# uname -n;date;ike cmd=list
p750_0_vio2
Tue Sep 16 11:35:17 EDT 2014
Phase Tun Id Status Local Id Remote Id
1 1 Active 172.16.22.5 172.16.22.10
2 1 Active 172.16.22.5-172.16.22.5 172.16.22.10-172.16.22.10

hscroot@hmc7:~>date;ls|parmigr -r lpar -m p750_2|grep LPAR_2
Wed Sep 17 11:42:10 EDT 2014
[1]+ Done migrlpar -o m -m p750_2 -t p750_0 -p LPAR_2

# uname -n;date;ike cmd=list
p750_2_vio1
Wed Sep 17 11:47:35 EDT 2014
No tunnels match your request.

# uname -n;date;ike cmd=list
p750_0_vio2
Tue Sep 16 11:47:02 EDT 2014
No tunnels match your request.

hscroot@hmc7:~>ls|syscfg -r lpar -m p750_0 -F name,state
p750_0_powervc,Not Activated
p750_0_vio2,Running
p750_0_vio1,Running
LPAR_2,Running
```

4.13 PowerVM NovaLink

PowerVM NovaLink is one of the latest features of IBM PowerVM. It is a software interface for virtualization management that can be installed on IBM POWER8 systems. PowerVM NovaLink can be installed in a logical partition running Ubuntu Linux.

The PowerVM NovaLink software is delivered using standard Debian packages that can be installed, removed, and updated just like any other software in Ubuntu Linux.

The PowerVM NovaLink stack consists of the following components:

- ▶ PowerVM NovaLink Core Services. These services provide direct interfaces to the managed system.
- ▶ OpenStack Services. These services provide drivers and plug-ins for use by OpenStack-based management solutions, including PowerVC.

PowerVM NovaLink Core Services interact with PowerVM by using the following interfaces:

- ▶ REST API. This API defines PowerVM REST objects and Python-based API wrappers for interactions with IBM PowerVM-based systems. The wrappers allow for retrieving and setting individual attributes of a REST object. The REST API is similar to that used on the HMC.
- ▶ CLI. This command-line interface (CLI) is a Python-based interface used for administrative functions on an IBM PowerVM-based system and allows for shell interaction with PowerVM. This CLI differs from that of the HMC and encompasses both Hypervisor and VIOS configurations.

The security model of PowerVM NovaLink is similar to the security model of the VIOS. PowerVM NovaLink CLI has a restrictive CLI used for management and administrative operations. It is not intended for regular users of the Ubuntu Linux logical partition.

The CLI uses the **pvmctl** command for most operations. The **pvmctl** command can be run only by users who are in the `pvm_admin` group. This is similar to the VIOS CLI where command-line utilities are accessible to only the user `padmin`.

The **pvmctl** command accepts only a restricted set of specific subcommands. Each subcommand is performed against a specific object and each object is associated with a specific list of subcommands.

The following object types are supported by the PowerVM NovaLink CLI:

- ▶ ManagedSystem (sys)
- ▶ LogicalPartition (lpar or vm)
- ▶ VirtualIOServer (vios)
- ▶ SharedStoragePool (ssp)
- ▶ IOSlot (io)
- ▶ LoadGroup (lgrp)
- ▶ LogicalUnit (lu)
- ▶ LogicalVolume (lv)
- ▶ NetworkBridge (nbr or bridge)
- ▶ PhysicalVolume (pv)
- ▶ SharedEthernetAdapter (sea)
- ▶ SharedStoragePool (ssp)
- ▶ VirtualEthernetAdapter (vea or eth)
- ▶ VirtualFibreChannelMapping (vfc or vfcmapping)
- ▶ VirtualMediaRepository (vmr or repo)
- ▶ VirtualNetwork (vnet or net)
- ▶ VirtualOpticalMedia (vom or media)
- ▶ VirtualSCSIMapping (scsi or scsimapping)
- ▶ VirtualSwitch (vswitch or vsw)

When integrating with VIOS, PowerVM NovaLink uses the **viosvrmd** command, which allows a legitimate user to run commands against a VIOS in a manner similar to the HMC **viosvrmd** command.

For a more comprehensive description of the various modes for accessing memory, see the following web page:

<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Power%20Systems/page/Introducing%20PowerVM%20NovaLink>

4.14 Conclusion

This chapter introduces the core concepts of IBM PowerVM and explains how they relate to security of IBM Power Systems environments. It also explains why security in such environments is a logical consequence of design and features employed by PowerVM.



IBM PowerKVM security

This chapter describes how to manage IBM PowerKVM security components to provide a safe virtual environment and to ensure reliability in your environment. An overview of key security features that are related to virtualized environments is provided and also suggestions for the implementation of cloud security.

The following topics are covered in this chapter:

- ▶ 5.1, “PowerKVM architecture overview” on page 88
- ▶ 5.2, “PowerKVM security considerations” on page 92
- ▶ 5.3, “Conclusion” on page 136

5.1 PowerKVM architecture overview

IBM PowerKVM is an important element of the IBM open cloud computing strategy. This software can help with the adoption and deployment of infrastructure as a service (IaaS) offerings. The IBM Power Systems provide an open virtualization choice for Linux systems based on the POWER8 technology.

Note: This chapter describes IBM PowerKVM version 3.1 that was available in November 2015.

PowerKVM is an open-source hypervisor that can efficiently and effectively run Red Hat Enterprise Linux, SUSE Linux Enterprise Server, and Ubuntu operating systems in virtual machines, providing strong performance, scalability, and security for the Linux environment.

The virtualization technology enables you to share real compute, memory, and I/O resources through server virtualization. These virtual resources are used by virtual machines that run on the PowerKVM virtualized server.

Figure 5-1 shows the PowerKVM infrastructure layout.

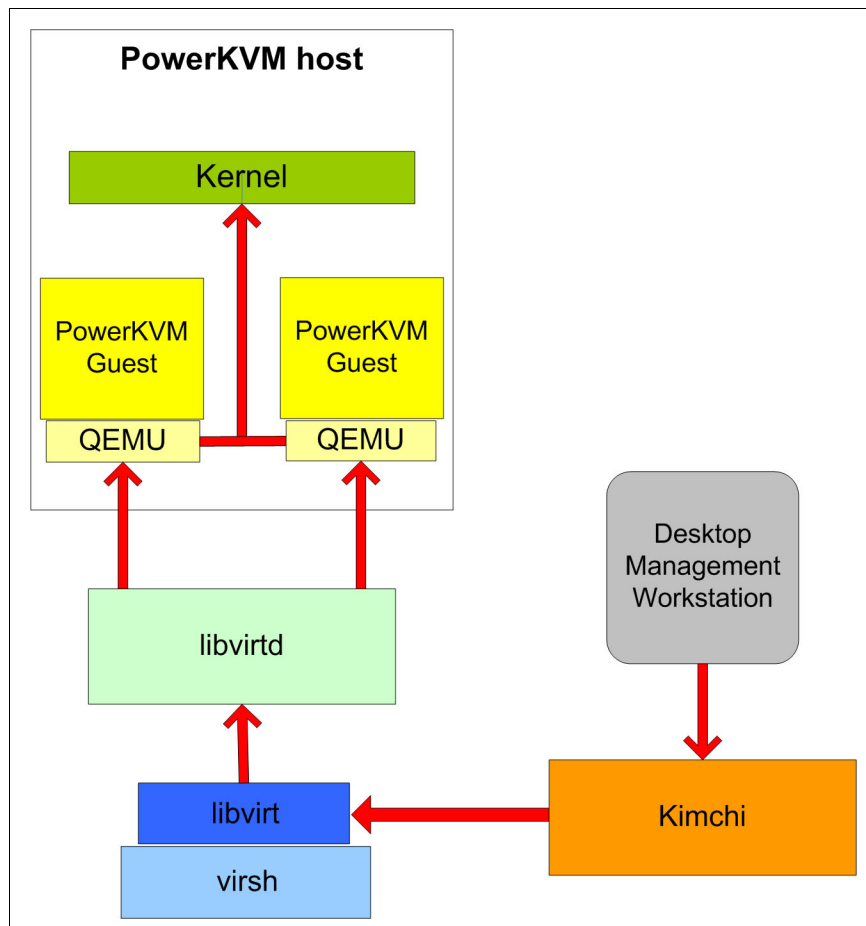


Figure 5-1 PowerKVM infrastructure layout diagram

The following PowerKVM components are described in this section:

- ▶ 5.1.1, “PowerKVM host” on page 89
- ▶ 5.1.2, “PowerKVM guest” on page 89
- ▶ 5.1.3, “Quick Emulator (QEMU)” on page 89
- ▶ 5.1.4, “The libvirt library” on page 89
- ▶ 5.1.5, “The virsh virtualization shell tool” on page 90
- ▶ 5.1.6, “Kimchi” on page 91

5.1.1 PowerKVM host

The PowerKVM host is a physical server running a KVM hypervisor that allows multiple Linux operating systems to share a single POWER8 server. All management of the PowerKVM guests is performed in this server, interacting directly with hypervisor.

You can access the PowerKVM host from a desktop management station by using one of the authentication methods listed in 5.2.1, “Authentication” on page 93.

5.1.2 PowerKVM guest

PowerKVM guests are virtual machines that are running on PowerKVM host. You can access them directly through the SSH command-line or by using a VNC client to log in to graphical remote console.

The following Linux distributions are supported for PowerKVM:

- ▶ Red Hat Enterprise Linux Version 6.5 or later
<http://www.redhat.com/en/technologies/linux-platforms/enterprise-linux>
- ▶ SUSE Linux Enterprise SLES 11 SP3 or later
<https://www.suse.com/products/server>
- ▶ Ubuntu 14.04 or later
<http://www.ubuntu.com/server>

5.1.3 Quick Emulator (QEMU)

The Quick Emulator (QEMU) is another important part of the PowerKVM infrastructure. When used with PowerKVM, QEMU primarily provides the virtualization mechanism and does basic operations such as the following tasks:

- ▶ Separate virtual machines instances from each other.
- ▶ Create disk image files.
- ▶ Perform I/O management between PowerKVM host and guests.

QEMU also works as an emulator, but this is not covered in this book.

5.1.4 The libvirt library

The *libvirt* library provides a common and stable layer sufficient to securely manage PowerKVM on a POWER8 server locally or remotely. It provides a stable, consistent API for management across a variety of virtualization technologies and all machine lifecycle events.

Currently, libvirt uses XML-based configuration files (located under `/etc/libvirtd` and `/var/lib/libvirt`) to define the virtualized hardware. Each PowerKVM guest has a specific XML file containing virtual devices assigned and available for utilization.

The libvirt library is used by the libvirtd daemon to communicate with the virtualization system. It is also used by the management tools. It runs on the PowerKVM host server on which the libvirt tools are started.

Several applications are being successfully built on libvirt. One of these applications is `virsh`, which is a virtualization shell.

5.1.5 The `virsh` virtualization shell tool

The `virsh` command-line tool is for management of PowerKVM guests. It is heavily dependent on the libvirt library. This shell permits use of much of the libvirt functionality. It provides capabilities to create, edit, or delete PowerKVM guests, manage storage pools, memory and processor use, and migrate virtual machines between PowerKVM hosts.

Note: Like most other Linux commands, the `virsh` commands can be run as a background process. However, a preferred practice is to avoid running `virsh` in this mode because unpredictable errors and timeouts can occur during the command lifecycle.

Table 5-1 lists several common `virsh` command options with descriptions.

Table 5-1 Common `virsh` command options

Command option	Description
<code>virsh connect</code>	Connect to the PowerKVM hypervisor.
<code>virsh create xmlfile.xml</code>	Create and start a guest from an XML configuration file.
<code>virsh list --all</code>	List all the guests on a host.
<code>virsh dumpxml guest_name</code>	Create an XML configuration file of the guest as an output file.
<code>virsh start guest_name</code>	Start an inactive guest.
<code>virsh destroy guest_name</code>	Immediately stop the guest.
<code>virsh define xmlfile.xml</code>	Create a guest from an XML configuration file. The guest is not started.
<code>virsh reboot guest_name</code>	Restart the guest.
<code>virsh restore fileName</code>	Restore a guest from a saved file.
<code>virsh resume guest_name</code>	Resume a guest that was paused.
<code>virsh save guest_name fileName</code>	Save the state of the guest to a file.
<code>virsh suspend guest_name</code>	Pause the guest.
<code>virsh undefine guest_name</code>	Delete the guest, but not the image file.
<code>virsh nodeinfo</code>	Display information about the host.

For more information about using `virsh`, see the Virsh Command Reference:

<http://libvirt.org/sources/virshcmdref/html>

5.1.6 Kimchi

Kimchi is a package for PowerKVM that allows an administrator to manage a single system graphically through a web interface. Kimchi works with libvirt to provide access for management of the I/O resources and virtual machines. Kimchi runs as a daemon and is started by default for preloaded PowerKVM systems.

Kimchi provides a graphical interface to do the following tasks:

- ▶ Create operating system templates
- ▶ Manage storage pools
- ▶ Manage virtual network devices
- ▶ Create virtual PowerKVM guests
- ▶ Backup the PowerKVM configuration
- ▶ Manage firmware upgrades
- ▶ Manage system health checks

Kimchi uses Pluggable Authentication Modules (PAM) to authenticate users. The two options to log in to Kimchi are as follows:

- ▶ Root access: Use the root user ID to manage Kimchi.
- ▶ Non-root access: Create a new user ID, removing access to the PowerKVM host and providing sudo root access for all commands in the `/etc/sudoers` file as shown in Example 5-1.

Example 5-1 Creating a user ID with Kimchi management access only

```
[root@powerkvm7 /]# useradd -g ibmadmin -c "Non-root userid for Kimchi
administration" -s /bin/false kimchi
```

```
[root@powerkvm7 log]# passwd kimchi
Changing password for user kimchi.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

```
[root@powerkvm7 /]# visudo
## Add line below for root sudo access of kimchi user id
kimchi ALL=(ALL) NOPASSWD: ALL
```

Preferred practice: It is considered a preferred practice to not use the `root` user ID for Kimchi management. Non-root user IDs with `/bin/false` for remote login are suggested for this access as shown in Example 5-1.

To connect to the Kimchi management interface, use the following URL format, where `PowerKVM_host` represents the fully qualified domain name (FQDN) or Internet Protocol (IP) address of the PowerKVM host server:

`https://PowerKVM_host:8001`

The following URL is an example:

`https://192.168.100.99:8001`

Figure 5-2 depicts the initial panel with PowerKVM guests running or stopped.

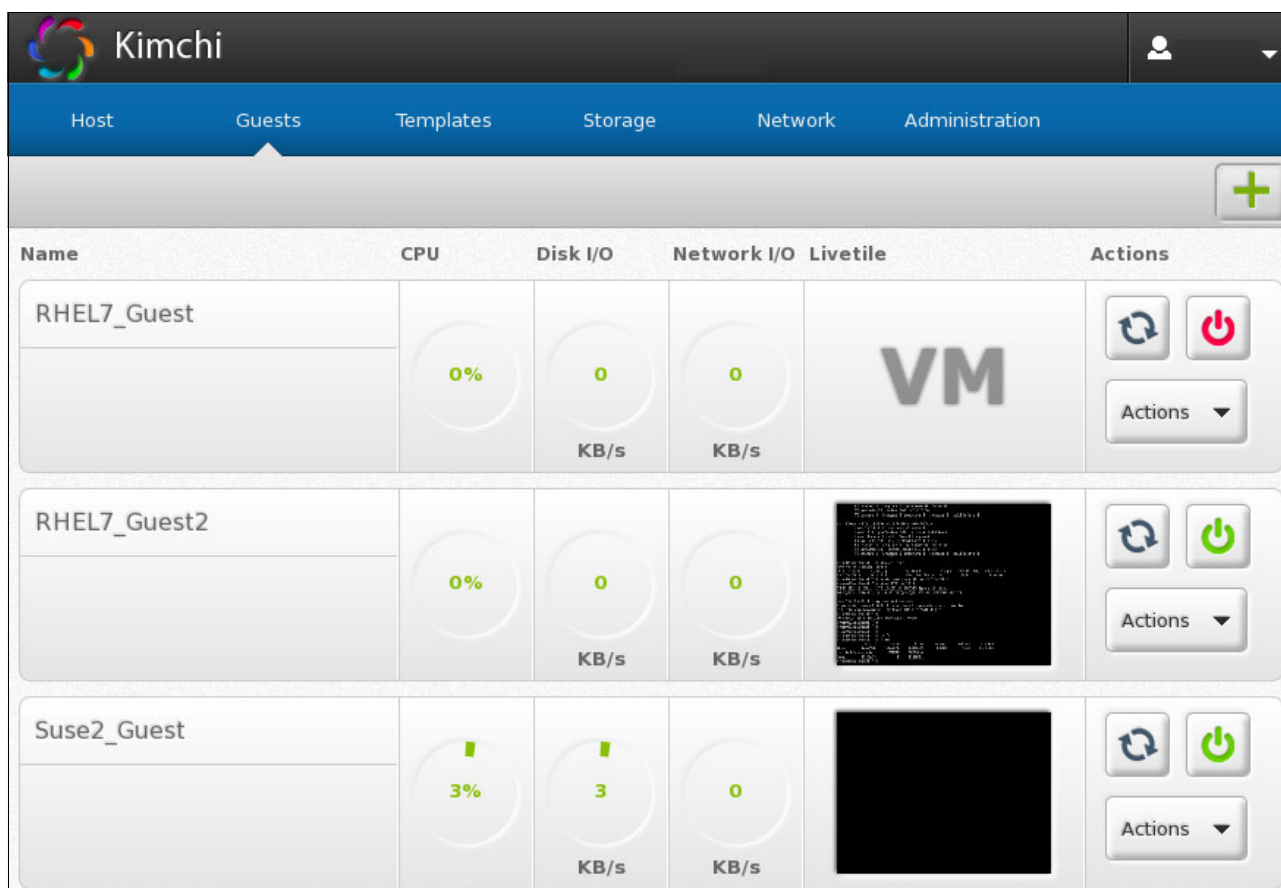


Figure 5-2 Kimchi initial panel

For more information about Kimchi, see the following website:

<https://github.com/kimchi-project/kimchi>

5.2 PowerKVM security considerations

PowerKVM is ready to be used by an IT administrator after the installation process. You can create storage locations, virtual networks, operating system (OS) templates, and deploy Linux virtual machines running Red Hat, SUSE, or Ubuntu.

This section reviews important aspects to enhance security capabilities and provide a stable and safe environment for organizations and users. The following topics are covered:

- ▶ 5.2.1, “Authentication” on page 93
- ▶ 5.2.2, “Networking” on page 103
- ▶ 5.2.3, “Firewall functionality with firewalld and iptables” on page 116
- ▶ 5.2.4, “Network filter driver” on page 117
- ▶ 5.2.5, “The sVirt service” on page 120
- ▶ 5.2.6, “Audit” on page 125
- ▶ 5.2.7, “PowerKVM guest image encryption” on page 129
- ▶ 5.2.8, “Guest live migration” on page 134

5.2.1 Authentication

This section describes different authentication methods and how they can be deployed and associated with additional security layers (such as remote management) by using the command-line interface. Preferred practices are highlighted regarding how authentication methods can be used to achieve a safe environment.

The following authentication methods are covered in this section:

- ▶ “Authentication using an SSH tunnel” on page 93
- ▶ “Authentication with PolicyKit” on page 93
- ▶ “Authentication with TLS and x509 certificates” on page 94
- ▶ “Authentication with SASL and TLS” on page 100
- ▶ “Authentication using VNC client” on page 100
- ▶ “Access to IPMItool and ASM on PowerKVM” on page 101

Note: The examples in this section use the `userlogin` login name.

Authentication using an SSH tunnel

The libvirt API, which is introduced in 5.1.5, “The `virsh` virtualization shell tool” on page 90, can perform various system management tasks through an SSH tunnel, including authentication. This means that management software that uses libvirt is also capable of using the authentication feature. In a standard PowerKVM deployment, this is the default authentication method.

When a root user logs in to a PowerKVM host through a standard SSH session, that KVM host does not require any extra configuration.

The `libvirtd` daemon that runs on a PowerKVM host regards connections that come through an SSH tunnel as though they are local. By using the SSH tunnel, you can remotely manage PowerKVM hosts by running the `virsh` command, or access your PowerKVM host graphical consoles by using the VNC viewer of your choice.

Example 5-2 shows how a root user authenticates to a PowerKVM host using an SSH tunnel.

Example 5-2 Root authentication using SSH tunnel

```
[root@oc5662730622 ~]# virsh -c qemu+ssh://root@powerkvm7/system
root@powerkvm7's password:
Welcome to virsh, the virtualization interactive terminal.
Type: 'help' for help with commands
      'quit' to quit
virsh # quit
```

The following sections, address other important topics concerning authentication, such as creating additional user IDs and groups, using certificate-based authentication, using MD5 encryption, and other options.

Authentication with PolicyKit

To allow additional users to access the KVM hosts, you must create those user IDs and manage them in optional groups.

The PolicyKit application framework provides functionality to manage user IDs and groups, and to access the PowerKVM host using an SSH tunnel.

The following Linux command creates a new user ID:

```
useradd -g polkit_admin -c "TolKit User Authentication" userlogin_polkit  
passwd userlogin_polkit
```

To grant the polkit_admin Linux group and all user IDs of this group access to the PowerKVM host, create the following PolicyKit configuration file:

/etc/polkit-1/localauthority/50-local.d/50-org.example-libvirt-remote-access.pkla

The content for this file is shown in Example 5-3.

Example 5-3 Content of the newly created .pkla file

```
[libvirt Management Access]  
Identity=unix-group:polkit_admin  
Action=org.libvirt.unix.manage  
ResultAny=yes  
ResultInactive=yes  
ResultActive=yes
```

After the .pkla file is created, restart the libvirtd and polkit daemon by using the following commands:

```
systemctl restart libvirtd  
systemctl restart polkit
```

To log in to the PowerKVM by using the userlogin_polkit user ID created, run the commands shown in Example 5-4.

Example 5-4 Login using userlogin_polkit user ID

```
virsh -c qemu+ssh://userlogin_polkit@powerkvm7/system  
userlogin_polkit@powerkvm7's password:  
Welcome to virsh, the virtualization interactive terminal.  
Type: 'help' for help with commands  
      'quit' to quit  
virsh # quit
```

Authentication with TLS and x509 certificates

Transport Layer Security (TLS) connections are made secure by using digital signature verification when users and servers exchange X.509 certificates that were previously signed by a recognized certificate authority (CA).

In other less common scenarios that need improved security, the server also requires the client to present a digitally signed certificate to prove its identity.

This section demonstrate how to create a local CA, use the local CA to digitally sign server and client certificates, and distribute the certificates for use. The **openssl** command is used to create private keys and certificates that can be used directly by libvirt.

Note: Based on experience of the authors, using this authentication option is considered to be the most secure and reliable.

Creating a CA key and certificate

You can create a CA key and certificate in your PowerKVM host. All certificates created are signed by a 2048-bit RSA key using an SHA256 hash algorithm. Use the following steps to create a CA key and certificate:

1. Log in to your PowerKVM host.
2. Create a temporary directory to store the files and change to that directory by using the following commands:

```
# mkdir cert_files  
# cd cert_files
```
3. Create a 2048-bit RSA key:

```
openssl genrsa -out cakey.pem 2048
```
4. Use the key to create a self-signed certificate to your local CA:

```
openssl req -new -x509 -days 1095 -key cakey.pem -out cacert.pem -sha256 -subj  
"/C=BR/L=Brazil/O=IBM/CN=my CA"
```
5. Check your CA certificate:

```
# openssl x509 -noout -text -in cacert.pem
```

Creating client and server keys

You can now create the client and server keys and certificates in your KVM host. To create both client and server certificates, use the following steps to create a certificate-signing request:

1. Create the keys by using the following commands:

```
# openssl genrsa -out serverkey.pem 2048  
# openssl genrsa -out clientkey.pem 2048
```
2. Create a certificate signing request for the server. Remember to change the `kvmhost.company.org` address (used in the server certificate request) to the fully qualified domain name of your KVM host.

```
# openssl req -new -key serverkey.pem -out serverkey.csr \  
-subj "/C=US/O=IBM/CN=powerkvm7.ibm.com"
```
3. Create a certificate signing request for the client:

```
# openssl req -new -key clientkey.pem -out clientkey.csr \  
-subj "/C=BR/O=IBM/OU=virtualization/CN=root"
```
4. Create client and server certificates by using the following commands:

```
# openssl x509 -req -days 365 -in clientkey.csr -CA cacert.pem -CAkey cakey.pem  
-set_serial 1 -out clientcert.pem  
  
# openssl x509 -req -days 365 -in serverkey.csr -CA cacert.pem -CAkey cakey.pem  
-set_serial 94345 -out servercert.pem
```
5. Check the keys by using the following commands:

```
# openssl rsa -noout -text -in clientkey.pem  
# openssl rsa -noout -text -in serverkey.pem
```

Distributing the keys

When certificates and keys for both the server and the client are in a format that is readable by libvirt, distribute and configure them to be used by TLS. To distribute the keys and certificates to the server (PowerKVM host), use the following steps.

1. Copy the CA certificate file `cacert.pem` to the `/etc/pki/CA/cacert.pem` file:

```
# cp cacert.pem /etc/pki/CA/cacert.pem
```
2. As shown in Example 5-5, create the `/etc/pki/libvirt` directory and copy the `servercert.pem` server certificate file into it. Then, create the `/etc/pki/libvirt/private` directory and copy the `serverkey.pem` server key file into it. Be sure that only the root user is able to access the private key.

Watch your steps: If the keys or certificates are named incorrectly or copied to the wrong directories, the authorization fails.

Example 5-5 Creating directories and copying files into them

```
# mkdir /etc/pki/libvirt
# cp servercert.pem /etc/pki/libvirt/.
# mkdir /etc/pki/libvirt/private
# cp serverkey.pem /etc/pki/libvirt/private/.
# chmod -R o-rwx /etc/pki/libvirt/private
```

3. Verify that the files are placed correctly as shown in Example 5-6.

Example 5-6 Verifying the files are placed correctly

```
# find /etc/pki/CA/*|xargs ls -l
-rw-r--r-- 1 root root 821 Apr  9 15:10 /etc/pki/CA/cacert.pem
# ls -lR /etc/pki/libvirt
/etc/pki/libvirt:
total 16
drwxr-x--- 2 root root 4096 Apr  9 16:35 private
-rw-r--r-- 1 root root 751 Apr  9 15:11 servercert.pem

/etc/pki/libvirt/private:
total 8
-rw-r----- 1 root root 1040 Apr  9 15:11 serverkey.pem
```

For every configured desktop management station, repeat the following steps to place a copy of the client certificate (`clientcert.pem`) in the `/etc/pki/libvirt/` directory and the key (`clientkey.pem`) in the `/etc/pki/libvirt/private/` directory:

1. Log in to the desktop management station.
2. Copy the CA certificate (`cacert.pem`) from the PowerKVM host to the management station `/etc/pki/CA/` directory. Do not change the file name.

```
# scp powerkvm7.ibm.com:/etc/pki/CA/cacert.pem /etc/pki/CA/
```

3. As shown in Example 5-7, copy the client certificate (`clientcert.pem`) from the PowerKVM host to the `/etc/pki/libvirt/` directory and the client key (`clientkey.pem`) to the `/etc/pki/libvirt/private/` directory. Use the default file names. Be sure that only the root user is able to access the private key.

Watch your steps: If the keys or certificates are named incorrectly or copied to wrong directories, authorization fails.

Example 5-7 Copying files

```
# cp clientcert.pem /etc/pki/libvirt/.
# mkdir /etc/pki/libvirt/private
# cp clientkey.pem /etc/pki/libvirt/private/.
# chmod -R o-rwx /etc/pki/libvirt/private
```

4. Verify that the files are placed correctly as shown in Example 5-8.

Example 5-8 Verifying the files are placed correctly

```
# ls -lR /etc/pki/libvirt/
/etc/pki/libvirt/:
total 8
-rw-r--r-- 1 root root 767 2010-04-09 13:54 clientcert.pem
drwxr-xr-- 2 root root 4096 2010-04-09 14:00 private

/etc/pki/libvirt/private:
total 4
-rw-r--r-- 1 root root 1044 2010-04-09 13:55 clientkey.pem
```

Configuring the libvirt daemon

Be sure that the `libvirtd` daemon is listening to network connections and that the `libvirtd.conf` file specifies the allowed subjects and client certificates. To edit the `libvirtd` daemon configuration, complete the following steps:

1. Log in to the PowerKVM host.
2. Create a copy of the `/etc/sysconfig/libvirtd` file and the `/etc/libvirt/libvirtd.conf` file.
3. Edit the `/etc/sysconfig/libvirtd` file and ensure that the `--listen` argument is passed to the `libvirtd` daemon. This step ensures that the `libvirtd` daemon is listening to network connections. The following example shows the changes from the original file:

```
LIBVIRT_ARGS="--listen"
```

4. Edit the `/etc/libvirt/libvirtd.conf` file and configure a set of allowed subjects using the `tls_allowed_dn_list` directive in the `libvirtd.conf` file. The following example shows the changes from the original file:

```
+tls_allowed_dn_list = ["C=*,O=IBM,OU=virtualization,CN=*"]
```

This restricts acceptable client certificates to certificates with the `O=IBM,OU=virtualization` values, while the country (C) and common name (CN) might be assigned any value.

5. Restart the `libvirtd` daemon service for changes to take effect:

```
# systemctl restart libvirtd
Stopping libvirtd daemon: [ OK ]
Starting libvirtd daemon: [ OK ]
```

Allowing TLS-authenticated connection

Allow TLS-authenticated connections through the firewall of the PowerKVM host by opening its TCP port 16514.

To allow TLS authentication connections through the firewall of the PowerKVM host, create a new iptables firewall rule to permit connections from anywhere to 16514/TCP port:

```
firewall-cmd --permanent --zone=public --add-port=16514/tcp
systemctl restart firewalld
```

For more information about firewalld, see 5.2.3, “Firewall functionality with firewalld and iptables” on page 116.

Verification

Verify that remote management is working from the desktop management station by running the **virsh** command as shown in Example 5-9.

Example 5-9 Verifying the remote management is working from the desktop management station

```
virsh -c qemu+tls://powerkvm7.ibm.com/system
Please enter your authentication name: userlogin
Please enter your password:
Welcome to virsh, the virtualization interactive terminal.
Type: 'help' for help with commands
      'quit' to quit
virsh # quit
```

For more information about the **openssl** command, see the following website:

<http://www.openssl.org/docs/apps/openssl.html>

Authentication with SASL

Simple Authentication and Security Layer (SASL) provides user ID and password authentication with data encryption (Digest-MD5, by default). It maintains its own user ID database. User IDs are not required to exist on the PowerKVM host.

Create a new user ID and verify that it was created as expected by using the following commands:

```
saslpaswd2 -a libvirt userlogin_sasl
sasldblistusers2 -f /etc/libvirt/passwd.db
```

To enable SASL authentication, change three values in the `/etc/libvirt/libvirtd.conf` file as shown in Example 5-10.

Example 5-10 Editing the `/etc/libvirt/libvirtd.conf` file to enable SASL

```
listen_tls = 0          # Disable the listen_tls flag
listen_tcp = 1          # Enable the listen_tcp flag
auth_tcp = "sasl"       # Set the authentication scheme
```

After the libvirt daemon is configured, remove the comment of the Digest-MD5 mechanism in the `/etc/sasl2/libvirt.conf` configuration file as shown in Example 5-11.

Important: Example 5-11 is provide to illustrate the possibilities of using SASL and MD5. However, MD5 hashes are now considered unsafe and should not be used in a production environment.

Example 5-11 Digest-MD5 encryption configuration

```
# Default to a simple username+password mechanism
mech_list: digest-md5

# Before you can use GSSAPI, you need a service principle on the
# KDC server for libvirt, and that to be exported to the keytab file listed below
#mech_list: gssapi
#
# You can also list many mechanisms at once, then the user can choose
# by adding '?auth=sasl.gssapi' to their libvirt URI, eg
# qemu+tcp://hostname/system?auth=sasl.gssapi
#mech_list: digest-md5 gssapi
```

SASL uses the 16509/TCP port for authentication and by default it is closed. To enable it, uncomment the following line in the `/etc/sysconfig/libvirtd` file and restart the libvirt daemon:

```
LIBVIRTD_ARGS="--listen" # Make libvirtd listen for TCP/IP connections
systemctl restart libvirtd
```

PowerKVM does not accept connections to 16509/TCP port by default. You must create a new iptables firewall rule to permit access for users to the PowerKVM host using this port. Run the following commands to create the rule:

```
firewall-cmd --permanent --zone=public --add-port=16509/tcp
systemctl restart firewalld
```

More information: For more information about firewalld, see 5.2.3, “Firewall functionality with firewalld and iptables” on page 116.

To validate SASL authentication access, run the commands shown in Example 5-12.

Example 5-12 Logging in using SASL authentication

```
virsh -c qemu+tcp://userlogin_sasl@powerkvm7/system
userlogin_sasl@powerkvm7's password:
Welcome to virsh, the virtualization interactive terminal.
Type: 'help' for help with commands
      'quit' to quit
virsh # quit
```

Authentication with SASL and TLS

SASL authentication on TLS provides a new security layer for TLS authentication. You can use TLS encryption and certificates by additionally controlling user ID access on the PowerKVM host side.

To configure this kind of authentication, complete the following steps:

1. Configure TLS as shown in “Authentication with TLS and x509 certificates” on page 94.
2. Complete the configuration steps. First, enable SASL for TCP and TLS connections in the `/etc/libvirt/libvirtd.conf` configuration file and restart the libvirtd daemon as demonstrated in Example 5-13.

Example 5-13 Enabling SASL and TCP connections

```
[root@powerkvm7 ~]# egrep 'auth_tcp|auth_tls' /etc/libvirt/libvirtd.conf
auth_tcp = "sasl"
auth_tls = "sasl"
[root@powerkvm7 ~]# systemctl restart libvirtd
Redirecting to /bin/systemctl restart libvirtd.service
[root@powerkvm7~]#
```

3. Because you are using SASL on top of TLS, you can turn off SASL encryption to avoid additional overhead as TLS connections are already encrypted. You must comment out the `mech_list` variable in the `/etc/sasl2/libvirt.conf` file, as shown in the following example. Note for SASL authentication only this value must be kept activated.

```
# Default to a simple username+password mechanism
# mech_list: digest-md5 ## Comment out this encryption line
```

4. Review Example 5-14, which demonstrates the remote login using SASL on top of TLS.

Example 5-14 Login using SASL and TLS authentication

```
[root@oc5662730622 libvirt]# virsh -c qemu+tls://powerkvm7.ibm.com/system
Please enter your authentication name: userlogin_sasl
Please enter your password:
Welcome to virsh, the virtualization interactive terminal.
Type: 'help' for help with commands
      'quit' to quit
virsh #
```

Authentication using VNC client

SSH tunneling for VNC is used to remotely access a VNC server by using an encrypted tunnel. To access the virtual machine console by tunneling, use the locally created VNC port and any VNC client.

This section shows the use of SSH tunneling to ensure that management traffic is safe.

Example 5-15 verifies how to identify the VNC port that is assigned for a PowerKVM guest, in this example *Port 1*.

Example 5-15 Verifying which VNC port is assigned to a PowerKVM guest

```
virsh # list
  Id      Name                               State
-----
  4      PowerKVM_Guest_0                   running
  5      PowerKVM_Guest_1                   running

virsh # vncdisplay PowerKVM_Guest_1
:1
virsh # exit
```

In the standardization for the VNC port that is used for PowerKVM guests, use the following default:

590 + VNC port assigned

In this specific example, the VNC port of our PowerKVM_Guest_1 is 5901.

To create an SSH tunnel and redirect all traffic from 5901/TCP to an encrypted 5910/TCP port, run the following command:

```
ssh -f -N -L 5910:localhost:5901 userlogin@powerkvm7
```

Access your VNC client and create a new connection; use the following values:

Server localhost - 127.0.0.1
Port 5910

Suggestions:

- ▶ If you are using a remote authentication command line for your PowerKVM host management, a preferred practice is to use the TLS method. This method is reliable and you can use proper user ID management for accessing your PowerKVM host.
- ▶ If you have several PowerKVM hosts and LDAP server running in organization, consider using a single sign-on method for authentication. This approach is explained in Chapter 6, “IBM PowerVC security” on page 137.

Access to IPMITool and ASM on PowerKVM

Flexible service processor (FSP) is a equivalent of Integrated management module (IMM) in x86 and PowerKVM terminology. The FSP is a service processor that provides function to initialize the system chip sets, boot to a hypervisor, and access to PowerKVM firmware (OPAL). Using FSP, users are able to initially install the PowerKVM environment and provide some routine low level tasks like physically shutdown server, check server status, and so on.

The users can reach the FSP of Power Systems to manage through the following options:

- ▶ Advanced System Management (ASM)
- ▶ Intelligent Platform Management Interface (IPMI)

For more information about ASM and IPMI, see the Redbooks publication, *IBM PowerKVM Configuration and Use*, SG24-8231.

Advanced System Management (ASM)

Advanced System Management (ASM) is a Power server embedded GUI web interface. Using ASM, users are able to manage FSP directly from a web browser. Users can access the ASM from the any FSP by using the following URL:

`https://<fsp_ip_address>`

Figure 5-3 shows the PowerKVM ASM default page.

Note: The default ASM user is **admin** with the default password of **admin**. It is considered a preferred practice that the password be changed after installation of the Power server.



Figure 5-3 ASM default page

Intelligent Platform Management Interface (IPMI)

Intelligent Platform Management Interface (IPMI) is a communication interface and a software implementation that help you to manage the hardware. It is a layer below the operating system, so the system continues to be manageable even if the machine does not have an operating system installed. It is also used to get the console during initial set up of the machine. On an IBM Power system, the IPMI server is hosted in the service processor controller, so users have access to the IPMI using the FSP IP address.

The IPMITool is the default tool to use as console with PowerKVM. The IPMITool utility is available with most distributions of Linux. If you do not have a version of IPMITool installed on your system, you can download a version from the Source Forge website:

<http://ipmitool.sourceforge.net/>

Common IPMITool examples are shown in Table 5-2 on page 103.

Table 5-2 Common IPMITool commands

Command option	Description
ipmitool -I lanplus -H myserver.example.com -P mypass chassis power on	Powers on the server.
ipmitool -I lanplus -H myserver.example.com -P mypass chassis power off	Powers off the server.
ipmitool -I lanplus -H myserver.example.com -P mypass chassis status	Checks the server status.
ipmitool -I lanplus -H myserver.example.com -P mypass chassis power cycle	Power cycle the server.
ipmitool -I lanplus -H myserver.example.com -P mypass sel list	Returns an error log.
ipmitool -I lanplus -H myserver.example.com -P mypass fru print	Prints the FRU information.
ipmitool -I lanplus -H myserver.example.com -P mypass user list	Lists the IPMI users.

Note: The IPMI password should be changed after installation of the Power server.

Your IPMI password is set on the IBM PowerKVM host through the ASM menus. To change your IPMI password, use the following steps:

1. Access the ASMI menu.
2. From the main menu, select **Login Profile** → **Change Passwords**.
3. Select **IPMI** from the list of user IDs.
4. Enter the current password for the administrator and then enter and confirm a new password for IPMI. Click **Continue**.

For more information about ASM and IPMI, see the Redbooks publication, *IBM PowerKVM Configuration and Use*, SG24-8231.

LC models family: On 5 October 2015, IBM announced three new POWER8 based servers that are supported by PowerKVM. The list of models includes S812LC, S822LC with integrated NVidia K40 GPU capabilities, and S822LC with non-GPU. These models support only the service processor named baseboard management controller (BMC) and not the FSM. The BMC service processor provides a hypervisor and operating system-independent layer that uses the extended error detection and auto-recovering functions that are built into the POWER8 processor and memory modules. BMC also monitors the operation of the firmware during the boot process and monitors the hypervisor for termination. BMC supports the Intelligent Platform Management Interface (IPMI 2.0) and Data Center Management Interface (DCMI 1.5) for system monitoring and management. Security setup and configuration steps similar to the IPMI-tool process for FSP service processor-based server with the only one difference: the user name is ADMIN not admin. For future information about LC models, see *IBM Power Systems S812LC Technical Overview and Introduction*, REDP-5284.

5.2.2 Networking

This section addresses the following networking topics:

- ▶ “Architecture” on page 104
- ▶ “Bonding” on page 109
- ▶ “VLAN segmentation” on page 111
- ▶ “Open vSwitch” on page 114
- ▶ “Quality of service” on page 116

Architecture

The PowerKVM network architecture is designed to provide a means of communication between virtual machines. Most network options can be implemented with the **virsh** command.

Table 5-3 lists common network options available in PowerKVM using the **virsh** command.

Table 5-3 Common virsh network commands

Command option	Description
virsh net-info network_name	Returns information about a network.
virsh net-list	Returns a list of active networks. Can also add --all to return all defined network, including inactive ones.
virsh net-start network_name	Starts a defined but inactive network.
virsh net-create xml_file	Creates a virtual network from an XML file.
virsh net-autostart network_name	Configures a network to automatically start.
virsh net-define xml_file	Defines a network from an XML file. The network is not started.
virsh net-dumpxml network_name	Creates an XML file of the virtual network information of a guest.

The following network types can be used with PowerKVM:

► Bridge

In bridge mode, a PowerKVM guest has access to external networks and the virtual machine. This configuration is the most commonly used for business scenarios. It requires a specific physical network adapter to be used by the hypervisor.

Example 5-16 shows the default configuration of a bridged virtual adapter for a PowerKVM guest.

Example 5-16 Bridge configuration for a PowerKVM guest

```
[root@powerkvm7 /]# virsh net-dumpxml Prod
<network connections='2'>
  <name>Prod</name>
  <uuid>68f8a663-90bb-4c75-969f-dcd69d9d3d0e</uuid>
  <forward mode='bridge' />
  <bridge name='brenP3p1s0f0' />
</network>

[root@powerkvm7 /]# virsh dumpxml PowerKVM_Guest_0
...
<interface type='network'>
  <mac address='52:54:00:54:42:f6' />
  <source network='Prod' />
  <model type='virtio' />
  <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x0' />
</interface>
...
```

► Network address translation (NAT)

In network address translation (NAT) mode, the PowerKVM guest uses an internal IP address that is not valid for the assigned VLAN. The guest can access other network resources, but other resources cannot access the guest. This configuration does not require a specific network adapter.

Example 5-17 shows the default configuration of a NAT virtual adapter for a PowerKVM guest.

Example 5-17 NAT configuration for a PowerKVM guest

```
[root@powerkvm7 /]# virsh net-dumpxml NAT
<network>
  <name>NAT</name>
  <uuid>82278717-e94a-4a41-833a-f970673f89da</uuid>
  <forward mode='nat'>
    <nat>
      <port start='1024' end='65535' />
    </nat>
  </forward>
  <bridge name='virbr2' stp='on' delay='0' />
  <mac address='52:54:00:af:9c:77' />
  <ip address='192.168.200.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.200.129' end='192.168.200.255' />
    </dhcp>
  </ip>
</network>

[root@powerkvm7 /]# virsh dumpxml PowerKVM_Guest_0
...
<interface type='network'>
  <mac address='52:54:00:fa:64:f6' />
  <source network='NAT' />
  <model type='virtio' />
  <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x0' />
</interface>
...
```

In this NAT configuration, the IP address 192.168.200.1 is assigned for virbr2. The PowerKVM guests uses an IP address of network 192.168.200.0/24, because the IP ranges from .129 to .255 are used for DHCP. You can remove the <dhcp> entry if not required.

► Isolated

In isolated mode, the PowerKVM guest uses an internal IP address. The guest cannot access external networks and can only access network resources that use the same virtual network device type and the same network. This configuration does not require a specific network adapter.

Example 5-18 shows the default configuration of an isolated virtual adapter for PowerKVM guests.

Example 5-18 Isolated configuration for a PowerKVM guest

```
[root@powerkvm7 /]# virsh dumpxml PowerKVM_Guest_0
...
<interface type='network'>
  <mac address='52:54:00:5c:ab:e5' />
  <source network='Isolated' />
  <target dev='vnet3' />
  <model type='virtio' />
  <alias name='net0' />
  <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x0' />
</interface>
...

[root@powerkvm7 /]# virsh net-dumpxml Isolated
<network connections='1'>
  <name>Isolated2</name>
  <uuid>d4a14f12-9699-4544-a6ef-3feab22557fe</uuid>
  <bridge name='virbr3' stp='on' delay='0' />
  <mac address='52:54:00:8c:0a:be' />
  <ip address='192.168.150.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.150.129' end='192.168.150.255' />
    </dhcp>
  </ip>
</network>
```

The isolated mode example shows that the virbr3 adapter uses IP address 192.168.150.1 and the PowerKVM guests can use network 192.168.150.0/24 to communicate between themselves and the PowerKVM host. A DHCP range is also configured, but you can remove it if it is not required.

► PCI pass-through

In PCI pass-through mode, the PowerKVM guest uses a physical network adapter assigned specifically for its use. This adapter cannot be used for other virtual machines.

Example 5-19 shows how to identify PCI network adapters that are available in the PowerKVM host. This example uses the PCI network adapter that is in the physical location 0003:01:00.3.

Example 5-19 Identifying PCI network adapters

```
[root@powerkvm7 /]# lspci | grep -i Ethernet
0003:01:00.0 Ethernet controller: Broadcom Corporation NetXtreme BCM5719 Gigabit
Ethernet PCIe (rev 01)
0003:01:00.1 Ethernet controller: Broadcom Corporation NetXtreme BCM5719 Gigabit
Ethernet PCIe (rev 01)
0003:01:00.2 Ethernet controller: Broadcom Corporation NetXtreme BCM5719 Gigabit
Ethernet PCIe (rev 01)
```


0003:01:00.3 Ethernet controller: Broadcom Corporation NetXtreme BCM5719 Gigabit Ethernet PCIe (rev 01)

```
[root@powerkvm7 /]# virsh nodedev-list | grep pci
```

```
...
pci_0003_00_00_0
pci_0003_01_00_0
pci_0003_01_00_1
pci_0003_01_00_2
pci_0003_01_00_3
...
```

```
[root@powerkvm7 /]# virsh nodedev-dumpxml pci_0003_01_00_3
```

```
<device>
  <name>pci_0003_01_00_3</name>
  <path>/sys/devices/pci0003:00/0003:00:00.0/0003:01:00.3</path>
  <parent>pci_0003_00_00_0</parent>
  <driver>
    <name>tg3</name>
  </driver>
  <capability type='pci'>
    <domain>3</domain>
    <bus>1</bus>
    <slot>0</slot>
    <function>3</function>
    <product id='0x1657'>NetXtreme BCM5719 Gigabit Ethernet PCIe</product>
    <vendor id='0x14e4'>Broadcom Corporation</vendor>
    <iommuGroup number='3'>
      <address domain='0x0003' bus='0x01' slot='0x00' function='0x0' />
      <address domain='0x0003' bus='0x01' slot='0x00' function='0x1' />
      <address domain='0x0003' bus='0x01' slot='0x00' function='0x2' />
      <address domain='0x0003' bus='0x01' slot='0x00' function='0x3' />
    </iommuGroup>
  </capability>
</device>
```

The PCI pass-through network interface must be configured for PowerKVM guest. The value of the *domain*, *bus*, and *slot* variables are used in the XML configuration file of the guest. The PCI network device must be detached from the PowerKVM host.

Example 5-20 shows the steps to configure the adapter.

Example 5-20 Configuring a PCI network adapter for a PowerKVM guest

```
[root@powerkvm7 /]# virsh nodedev-detach pci_0003_01_00_3
```

```
Device pci_0003_01_00_3 detached
```

```
[root@powerkvm7 /]# virsh edit PowerKVM_Guest_0
```

```
...
<hostdev mode='subsystem' type='pci' managed="yes">
  <source>
    <address domain='0x0003' bus='0x01' slot='0x03' function='0x00' />
  </source>
  <driver name='vfio' />
</hostdev>
...
```

Restart the PowerKVM guest to ensure that the physical network adapter was assigned appropriately by using the following commands:

```
virsh destroy PowerKVM_Guest_0  
virsh start PowerKVM_Guest_0
```

Example 5-21 shows the PCI network adapter assigned for PowerKVM guest.

Example 5-21 Verifying the PCI network adapter was assigned correctly

```
powerkvmguest0:~ # lspci  
0000:00:01.0 Ethernet controller: Red Hat, Inc Virtio network device  
0000:00:02.0 USB controller: Apple Inc. KeyLargo/Intrepid USB  
0001:00:00.0 Ethernet controller: Broadcom Corporation NetXtreme BCM5719  
Gigabit Ethernet PCIe (rev 01)
```

Figure 5-4 on page 109 shows an architectural overview of implementing the network types:

- ▶ *Guest 1* represents a bridge example, communicating with the network in both flows.
- ▶ *Guest 2* and *Guest 3* represent isolated examples, communicating between themselves and Bridge2.
- ▶ *Guest 4* represents a NAT example, communicating with the network in a unique flow.
- ▶ *Guest 5* represents a PCI pass-through example that communicates directly with the network after a physical network card was assigned to the virtual machine.

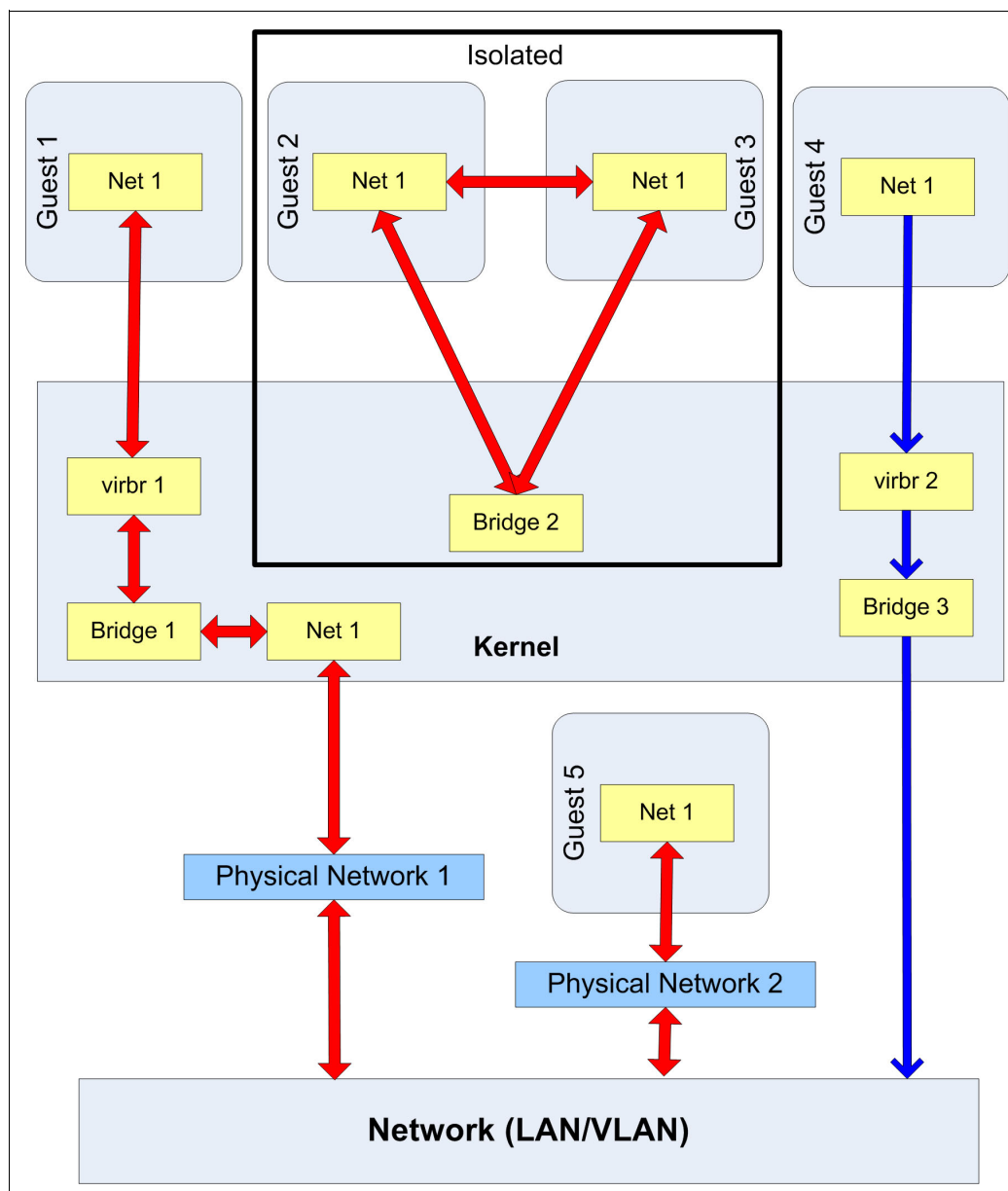


Figure 5-4 Network types available for PowerKVM

Bonding

The Linux bonding driver provides a method for aggregating multiple network interfaces into a single logical bonded interface. This bonded interface provides reliability because two or more interfaces are treated as one, creating a redundant network environment.

Figure 5-5 on page 110 depicts a network infrastructure with a bonded logical device created that provides access for VLAN10, VLAN11, and VLAN12 using bridges.

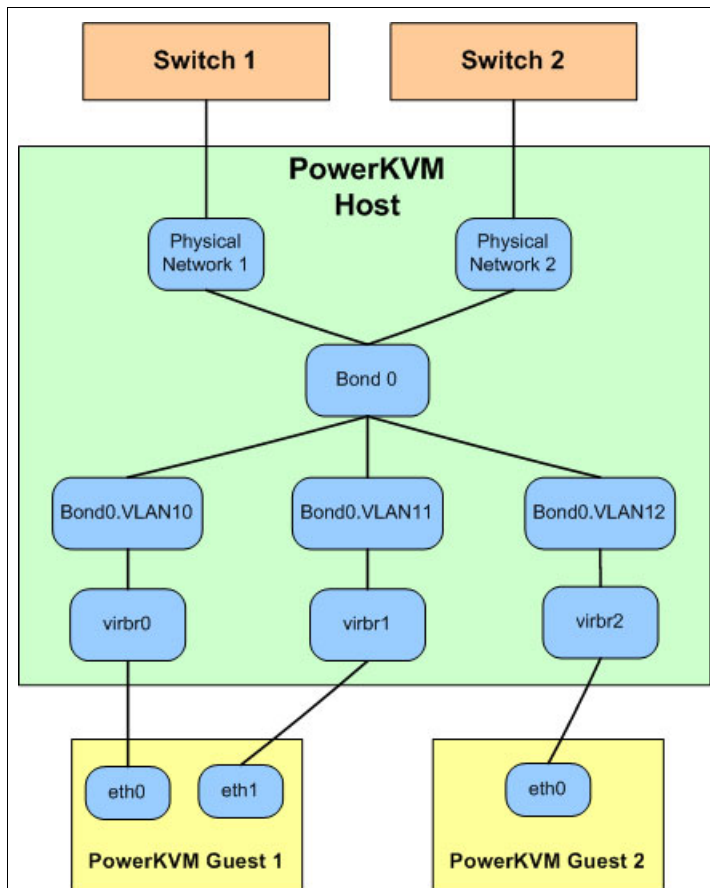


Figure 5-5 Bonding diagram

To configure bonding, complete the following steps:

1. Edit the network configuration files to redirect data to the bonded logical device. Example 5-22 shows editing the configuration file for the enP3p1s0f0 physical network interface.

Example 5-22 Editing the network configuration file for the enP3p1s0f0 interface

```

[root@powerkvm7 /]# vi /etc/sysconfig/network-scripts/ifcfg-enP3p1s0f0
NM_CONTROLLED=no
DEVICE=enP3p1s0f0
BOOTPROTO=None
HWADDR=6c:ae:8b:00:46:b4
TYPE=Ethernet
ONBOOT=yes
SLAVE=yes
MASTER=bond0

```

Do the same step for the second physical network interface, enP3p1s0f1 as shown in Example 5-23.

Example 5-23 Editing the network configuration file for the enP3p1s0f1 interface

```
[root@powerkvm7 /]# vi /etc/sysconfig/network-scripts/ifcfg-enP3p1s0f1
NM_CONTROLLED=no
DEVICE=enP3p1s0f1
BOOTPROTO=none
HWADDR=6c:ae:8b:00:46:b5
TYPE=Ethernet
ONBOOT=yes
SLAVE=yes
MASTER=bond0
```

2. Create a new network configuration file for the bond0 logical interface as shown in Example 5-24.

Example 5-24 Creating a new bond0 configuration file

```
[root@powerkvm7 /]# vi /etc/sysconfig/network-scripts/ifcfg-bond0
DEVICE=bond0
ONBOOT=yes
BOOTPROTO=none
BONDING_OPTS="mode=4 miimon=100"
MTU=9000
```

3. Restart the network services by using the following command:

```
systemctl restart network
```

You must use the *bond0* logical interface to create new bridges.

Note: Use the bonding method for networks with high priority to avoid any negative business impact. This way can prevent connectivity loss and provide a load balancer capability for the overall network throughput.

VLAN segmentation

The virtual LAN (VLAN) segmentation is a method to isolate network traffic within a physical network infrastructure. VLAN segmentation, also referred to as *802.1q tagging*, works by appending a tag identifying the VLAN ID to each TCP packet.

Most of the existing network infrastructures support 802.1q VLANs for complex environments. This method is a good configuration to reduce hardware cost because it can improve existing network access controls and can help prevent attacks to virtual machine guests from malicious users.

In addition, to increase the security of the PowerKVM host, you can configure one network interface for the host server and a separate network interface for the PowerKVM guests. In this configuration, the network traffic for the PowerKVM host works on a different subnet than the network traffic for the guest operating systems. This network configuration increases security resilience because it helps to isolate the PowerKVM host from the guest operating systems and prevents that malicious users gain unauthorized access to the hypervisor level.

PowerKVM offers 802.1q tagging by default. Complete the following steps to configure your VLAN segmentation in the PowerKVM host to be used for the PowerKVM guests:

1. Identify the VLAN IDs that will be assigned for PowerKVM guests.
2. Configure the external network infrastructure to allow traffic from those VLANs to the PowerKVM host:
 - a. Configure the network switch connected to the PowerKVM Host.
 - b. Configure the switch physical port that the PowerKVM host is connected to as a trunk (multiple VLANs) and a tagged (accepts tagged frames) port.
 - c. Allow traffic to the required VLAN IDs.

Single host only: If all of the PowerKVM guests reside on a single host, then step b and c are not required.

3. Create the virtual bridge in the KVM Host. Avoid mixing different VLANs in a single bridge.
4. Create a file called `ifcfg-<name>` in the `/etc/sysconfig/network-scripts` directory to create a permanent bridge configuration, where `<name>` is the bridge interface name.

Example 5-25 specifies a `br_v10` bridge with the following file:

`/etc/sysconfig/network-scripts/ifcfg-br_v10`

Example 5-25 Content of `/etc/sysconfig/network-scripts/ifcfg-br_v10` file

```
DEVICE=br_v10
TYPE=Bridge
BOOTPROTO=static
STP=yes
ONBOOT=yes
DELAY=0
```

Bridge availability: The `ONBOOT=yes` line assures that the bridge becomes available automatically after each boot. No IP address is required to this bridge and Spanning Tree Protocol (STP) is enabled.

5. If multiple guests are participating in the same VLAN ID (even if they use separate bridges), disable netfilter processing in bridging devices by appending the lines in Example 5-26 to the `/etc/sysctl.conf` file.

Example 5-26 Disabling netfilter processing in the `/etc/sysctl.conf` file

```
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
```

To reload the new kernel parameters, run the `sysctl` command:

```
sysctl -p
```

6. Configure one or more subinterfaces from the main, physical network interface (the trunk). Example 5-27 configures the subinterface eth0.10 that is assigned to VLAN ID 19. The file name is /etc/sysconfig/network-scripts/ifcfg-eth0.10.

Example 5-27 Subinterface en0.10 configuration

```
#VLAN 19 in trunk eth0
DEVICE=eth0.10
VLAN=yes
ONBOOT=yes
BRIDGE=br_v10
```

Naming:

- ▶ Renaming interfaces: The network interfaces are renamed to guarantee that each interface has the same name across reboots, no matter in which order they are activated. In Example 5-27, the eth1 network interface is renamed to enP3p1s0f0.
- ▶ Watch the naming convention: The value of DEVICE (which is the subinterface name) is in the form of <interface-name>.<VLAN-ID>, where <interface-name> is the name of the physical interface that this virtual bridge attaches to (enP3p1s0f0 in this example) and <VLAN ID> is directly taken from the subinterface name (number 10 in this example). Stripping the VLAN tags is optional.

7. Because ONBOOT is set to yes, the subinterface of the virtual bridge opens automatically on every reboot, however you can also start them manually by using the **ifup** command. The **brctl** command lists all virtual bridges and their assigned interfaces as shown in Example 5-28.

Example 5-28 Bringing up the subinterface of the virtual bridge

```
# ifup br_v10
# ifup enP3p1s0f0.10
# brctl show
bridge name      bridge id      STP enabled interfaces
br0      8000.00145ed87f4a  yes enP3p1s0f0
virbr0      8000.000000000000  yes
br_v10      8000.00145ed87f4a  yes enP3p1s0f0.10
```

8. When the bridge interface is running, you must adjust each PowerKVM guest configuration by assigning interfaces to their respective bridge or VLAN. Example 5-29 shows how to create a bridge interface for the PowerKVM guest.

Example 5-29 Creating a bridge interface in the PowerKVM guest

```
<interface type='bridge'>
<mac address='51:56:11:6c:1a:cd' />
<source bridge='br_v10' />
<target dev='vnet0' />
</interface>
```

9. Restart the modified guests so that the changes can take effect by using the following commands:

```
virsh destroy VM_Guest
virsh start VM_Guest
```

When the PowerKVM guest is online again, you can assign a specific IP address of VLAN ID 10 to the guest operating system for its network connection.

You can also configure 802.1q VLANs with the Kimchi interface, by completing the following steps:

1. Log into the Kimchi interface.
2. Select the Network window.
3. Create a network by clicking the green plus (+) icon.
4. Enter a name for your network.
5. Select **Bridged** as type.
6. Select **Enable VLAN** and enter a VLAN ID.
7. Click **Create**.

Open vSwitch

The PowerKVM host provides another solution for VLAN management. The Open vSwitch is an application that can create virtual switches using VLAN-tagged technology to isolate the traffic of a switch trunk port. For more information about the Open vSwitch, see the Open vSwitch website:

<http://openvswitch.org>

Complete the following steps to configure the Open vSwitch application to isolate traffic for different VLANs using a single VLAN ID per virtual network interface:

1. Start the Open vSwitch daemon by running the following command.

```
systemctl start openvswitch
```

Note: The **service** command is included in Red Hat Enterprise Linux 7 and PowerKVM for backward-compatibility. It has been replaced by the **systemctl** command, which is part of the system software. For details about the equivalence of these commands in RHEL releases, refer to the following website:

<https://access.redhat.com/articles/1189123>

You can check the status of your service by using the **systemctl status** command as shown in Example 5-30.

Example 5-30 Checking the status

```
systemctl status openvswitch
openvswitch.service - Open vSwitch
  Loaded: loaded (/usr/lib/systemd/system/openvswitch.service; enabled)
  Active: active (exited) since Mon 4s ago
  Process: 99513 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
ovs-vsctl show
368e3b-fa46-4e46-8fda-3410ec2c2284
ovs_version: "2.0.1"
```

2. Create a native Open vSwitch (OVS) bridge and add the physical interface to your bridge:

```
ovs-vsctl add-br br0
ovs-vsctl add-port br0 enP3p1s0f0
```

Renaming interfaces: The network interfaces are renamed to guarantee that each interface has the same name across reboots, no matter what order they are activated.

3. Create an XML file to configure the virtual network interface using VLAN ID 20 for virtual machines as shown in Example 5-31.

Example 5-31 Creating a new virtual network interface

```
[root@powerkvm7 ~]# vi /tmp/vlan20.xml
<network>
  <name>ovs-network</name>
  <forward mode='bridge' />
  <bridge name='br0' />
  <virtualport type='openvswitch' />
  <portgroup name='vlan-20'>
    <vlan>
      <tag id='20' />
    </vlan>
  </portgroup>
</network>
```

4. Configure and enable this interface for PowerKVM guest use as shown in Example 5-32.

Example 5-32 Configuring and enabling the XML file

```
[root@powerkvm7 ~]# virsh net-define vlan20.xml
Network ovs-network defined from vlan20.xml

[root@powerkvm7 ~]# virsh net-start ovs-network
Network ovs-network started

[root@powerkvm7 ~]# virsh net-autostart ovs-network
Network ovs-network marked as autostarted
```

5. Add the lines for the bridge ovs-network, created and configured for PowerKVM guest configuration as shown in Example 5-33.

Example 5-33 Adding new bridge ovs-network (VLAN 20) for guest.

```
[root@powerkvm7 ~]# virsh edit PowerKVM_Guest_0
...
<interface type='network'>
  <source network='ovs-network' />
  <target dev='vnet1' />
  <model type='virtio' />
</interface>
...
```

6. Restart the PowerKVM guest by using the following commands:

```
virsh destroy PowerKVM_Guest_0
virsh start PowerKVM_Guest_0
```

Quality of service

Quality of service (QoS) is an Internet standard that provides the feature of giving preferential benefits to certain types of IP traffic. PowerKVM version 3.3.1 supports QoS using the bandwidth element for networks that use forwarding like NAT. QoS is not supported for network bridges.

The bandwidth element uses inbound and outbound child elements to allow incoming and outgoing traffic to be set independently of each other. A user can have only one inbound and one outbound child element for each network interface. If you do not set this option, QoS is not presented. Also, if your bandwidth element specifies values for only incoming traffic, then outgoing traffic does not use QoS.

Example 5-34 provides the base XML syntax for the QoS feature in a network device.

Example 5-34 Base XML syntax for QoS feature in a network device

```
<forward mode='nat' dev='eth0' />
  <bandwidth>
    <inbound average='1000' peak='5000' burst='5120' />
    <outbound average='128' peak='256' burst='256' />
  </bandwidth>
```

5.2.3 Firewall functionality with firewalld and iptables

The PowerKVM host can be a potential denial-of-service (DoS) target. Typically, many virtual machines are hosted in a PowerKVM production environment, and an attack can potentially impact all guests. Several firewall rules are activated in the default installation of a PowerKVM host to avoid malicious access like a DoS attack.

By using the **nmap** command, shown in Example 5-35, you can list the open TCP ports that are accepted by the firewall configuration.

Example 5-35 TCP ports opened by default on the PowerKVM host

```
[root@oc5662730622 ~]# nmap powerkvm7

Starting Nmap 5.51 ( http://nmap.org ) at 2014-08-12 19:17 BRT
Nmap scan report for powerkvm7 (192.168.1.100)
Host is up (0.091s latency).
Other addresses for powerkvm7 (not scanned): 192.168.1.101
Not shown: 997 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
8000/tcp   open  http-alt
8001/tcp   open  vcom-tunnel

Nmap done: 1 IP address (1 host up) scanned in 11.22 seconds
```

The *firewalld* application provides a managed firewall with support for network and firewall zones to restrict and control access of network connections or interfaces. It supports IPv4, IPv6, and an interface for services or applications to add firewall rules directly.

The *firewalld* application works in conjunction with *iptables*. The *iptables* application allows the administration of the tables in the Linux kernel firewall. All firewall rules of the PowerKVM host are managed by *firewalld*.

When you require opening a TCP or UDP port for communication between external devices or desktop management (public) to the PowerKVM host, the **firewall** command is required as shown in Example 5-36.

Example 5-36 Firewall commands to open TCP ports

```
## Allowing SMTP (25/TCP) connections to PowerKVM host
firewall-cmd --permanent --zone=public --add-port=25/tcp
systemctl restart firewalld

## Allowing SNMP (161/UDP) connections to PowerKVM host
firewall-cmd --permanent --zone=public --add-port=161/udp
systemctl restart firewalld
```

Suggestion: Open a minimal set of network ports, allowing traffic for approved applications to enter and leave the network while blocking other network traffic.

5.2.4 Network filter driver

The network filter driver provides a configurable network filtering capability interacting directly with *ebtables*, *iptables*, and *ip6tables*. The filters are implemented as XML files that are managed by libvirt and are referenced by a libvirt object that requires their functionality.

Several useful default filters exist for PowerKVM deployed in the default installation. These include filters to avoid MAC, IP address, and ARP spoofing. *Spoofing* is an attack in which one malicious person or program successfully masquerades as another by falsifying data to obtain confidential data of your target.

All filters are listed in Example 5-37.

Example 5-37 Filters available in PowerKVM default installation

```
[root@powerkvm7 /]# virsh nwfilter-list
UUID                                Name
-----
0168c9db-7476-4bcc-8fe7-29b61cf9fb3f allow-arp
8e5b37c4-a766-4f83-b1e6-af158666daca allow-dhcp
25f76d1d-7567-4441-9fdf-25f99794de5a allow-dhcp-server
8d6ff7c7-bc6c-4747-bc78-8c64ff6252ea allow-incoming-ipv4
42c65dd0-1fc6-4ba4-a3f1-4d15192453ee allow-ipv4
c8e24a32-7294-40c7-8e7f-0126b5228a3f clean-traffic
99ea206c-846e-411c-be05-bee2319ecd97 no-arp-ip-spoofing
6fc62015-eb57-46ad-969d-59aa8e1a5a7b no-arp-mac-spoofing
e2e32cc1-25e2-4b70-8bf3-c2a45f8572b1 no-arp-spoofing
bc07a174-bb44-4875-8cb6-155a8ef08bc3 no-ip-multicast
f5a1b319-a5f4-474d-89bf-268052906e85 no-ip-spoofing
7afc7a51-0716-43ba-8a8d-dc6b853bd024 no-mac-broadcast
f9293efb-817c-4274-b173-a79074ea85f5 no-mac-spoofing
60e0ff42-2bd1-4a7b-ae3c-b0ec032ae6c6 no-other-l2-traffic
2d4717ff-a325-4f44-bda2-961384ba0e74 no-other-rarp-traffic
2f9c21b9-7a61-4d2d-86bc-97d8a41aa9a8 qemu-announce-self
f041ccab-4a38-4713-8724-8b0e279f343c qemu-announce-self-rarp

[root@powerkvm7 /]#
```

Suggestion: Use the *clean-traffic* filter when possible. It uses the most important available filters to keep your PowerKVM hosts and guests safe.

Example 5-38 shows the high level construct of a default filter created in XML mode.

Example 5-38 High level view of a filter

```
[root@powerkvm7 /]# virsh nwfilter-dumpxml XXXX
<filter name='NAME' chain='XXXX'>
  <uuid>d8367f2a2-6b48-0s12-9c61-ec2783536c27</uuid>

  <rule ...>
    ....
  </rule>

  <filterref filter='XXXX' />
</filter>
```

For more information about the network filter driver, see the following website:

<http://libvirt.org/formatnwfilter.html>

Important: The following section, “Blocking MAC and IP address spoofing”, demonstrates how to configure PowerKVM guests to block spoofing activities from malicious people.

Blocking MAC and IP address spoofing

The Media Access Control (MAC) is a physical address included in network interfaces for communications on a network segment. An Internet Protocol (IP) address is a numerical label assigned to each network device participating in a computer network that uses the IP for communication.

These addresses must be unique to avoid security breaches in the network or systems. This section demonstrates how to restrict malicious actions in the PowerKVM environment.

Example 5-39 confirms that a PowerKVM guest is communicating with other servers.

Example 5-39 Testing communication with other servers

```
[root@PowerKVMGuest1 /]# ping 172.16.20.1
PING 172.16.20.1 (172.16.20.1) 56(84) bytes of data.
64 bytes from 172.16.20.1: icmp_seq=1 ttl=64 time=0.252 ms
64 bytes from 172.16.20.1: icmp_seq=2 ttl=64 time=0.307 ms
64 bytes from 172.16.20.1: icmp_seq=3 ttl=64 time=0.265 ms
64 bytes from 172.16.20.1: icmp_seq=4 ttl=64 time=0.197 ms
64 bytes from 172.16.20.1: icmp_seq=5 ttl=64 time=0.229 ms
64 bytes from 172.16.20.1: icmp_seq=6 ttl=64 time=0.270 ms
--- 172.16.20.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 4998ms
rtt min/avg/max/mdev = 0.197/0.253/0.307/0.036 ms
[root@PowerKVMGuest1 ~]#
```

Edit the PowerKVM guest configuration to add the clean-traffic filter from the network filter driver. Add the filterref filter line as shown in Example 5-40.

Example 5-40 Editing the PowerKVM guest configuration file to add the filterref filter line

```
$ virsh edit PowerKVM_Guest_1
...
<interface type='network'>
  <mac address='52:54:00:54:42:f6' />
  <source network='Prod' />
  <model type='virtio' />
  <filterref filter='clean-traffic' />
...
</interface>
```

Prefixes in names: Create a target device name that does *not* start with the following prefixes, this is because those prefixes conflict with the libvirt automatic naming scheme and are ignored:

- ▶ vnet
- ▶ vif

These changes take effect after the next start up, therefore PowerKVM guests must be restarted now, using the following commands:

```
virsh destroy PowerKVM_Guest_1
virsh start PowerKVM_Guest_1
```

When the PowerKVM guest is running, you can do a spoofing test to ensure that the configuration was successfully implemented. Change the MAC address of the virtual network interface:

```
[root@PowerKVMGuest1 ~]# ifconfig eth0 hw ether 52:54:00:60:EF:FF
```

Verify whether the MAC address change was run as shown in Example 5-41.

Example 5-41 Checking the MAC address spoofed

```
[root@PowerKVMGuest1 /]# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.22.7 netmask 255.255.252.0 broadcast 172.16.23.255
    inet6 fe80::5054:ff:fe54:42f6 prefixlen 64 scopeid 0x20<link>
    ether 52:54:00:60:EF:FF txqueuelen 1000 (Ethernet)
    RX packets 176 bytes 16394 (16.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 63 bytes 10575 (10.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

The MAC address was successfully spoofed, so you can test connectivity now. The PowerKVM guest lost communication with the internal network, as shown in Example 5-42.

Example 5-42 Checking communication after MAC changed

```
[root@PowerKVMGuest1 /]# ping 172.16.20.1
PING 172.16.20.1 (172.16.20.1) 56(84) bytes of data.

--- 172.16.20.1 ping statistics ---
8 packets transmitted, 0 received, 100% packet loss, time 7002ms
```

Do the same test for an IP address, changing the IP address of the PowerKVM guest to another one as shown in Example 5-43.

Example 5-43 Changing the IP address and checking communication after the IP address is changed

```
[root@PowerKVMGuest1 ~]# ifconfig eth0 172.16.22.5 netmask 255.255.252.0 up
```

```
[root@PowerKVMGuest1 /]# ping 172.16.20.1
PING 172.16.20.1 (172.16.20.1) 56(84) bytes of data.
```

```
--- 172.16.20.1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3014ms
```

5.2.5 The sVirt service

The secure virtualization (sVirt) service is a Security Enhanced Linux (SELinux) framework included in libvirt and isolates virtual machines by using a flexible and customizable security policy. SELinux provides resource isolation and confinement for processes running on top of the Linux kernel, including virtual machine processes on the PowerKVM host. SELinux enforces the idea that programs should be limited to what files they can access and what actions they can run.

SELinux uses Mandatory Access Control (MAC) type for access control. The primary security model that is used by SELinux is Domain-Type Enforcement (DTE), the other model is Multi-Level Security (MLS). Secure virtualization or sVirt uses the simplest implementation of MLS to restrict the access of guest images and inter-process communication between differently labeled guests called Multi Category Security (MCS).

For more information about DTE and MLS and MCS models, see the following website:

http://selinuxproject.org/page/NB_MLS

The main objective of sVirt is to protect the PowerKVM hosts and guests from malicious attacks in the hypervisor, applying access policy across different processes as QEMU and files as disk images.

If SELinux and sVirt are disabled, a malicious person might potentially explore a vulnerability and access processes and files on other PowerKVM guests, exposing confidential data.

Figure 5-6 shows a PowerKVM host with SELinux disabled and PowerKVM Guest 2 accessing the disk image of PowerKVM Guest 1 after a vulnerability was exploited.

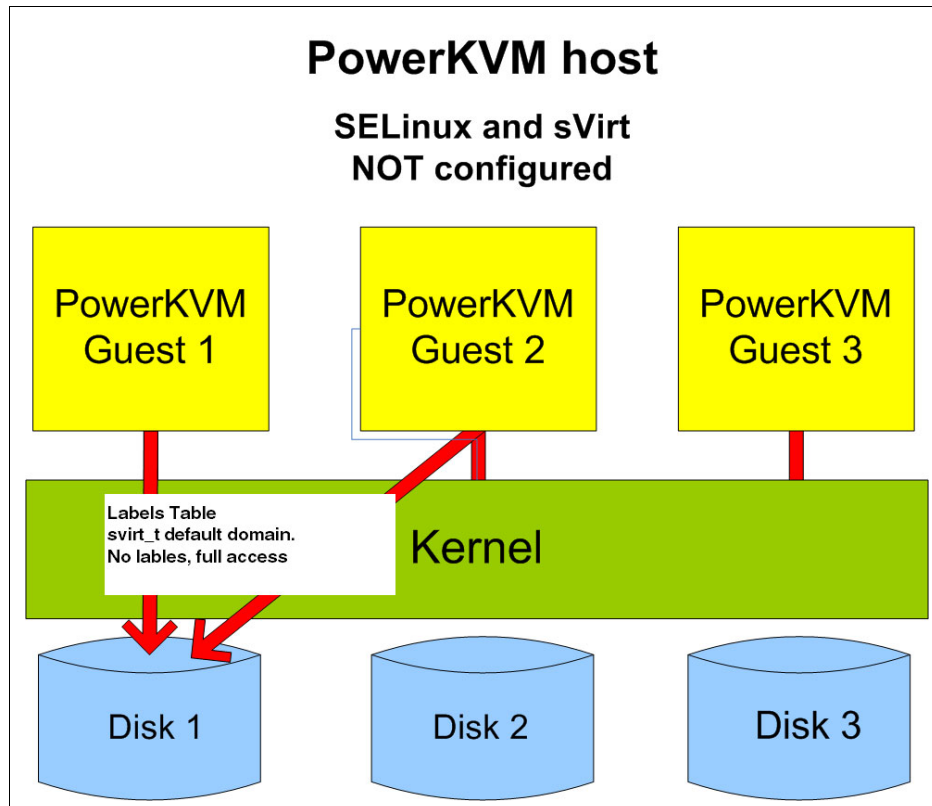


Figure 5-6 PowerKVM guest accessing disk image of another guest

By default, SELinux and sVirt are enabled for PowerKVM hosts. They provide security labels to ensure that PowerKVM guests can only access their own processes and files.

Figure 5-7 shows a PowerKVM host with SELinux and sVirt working properly.

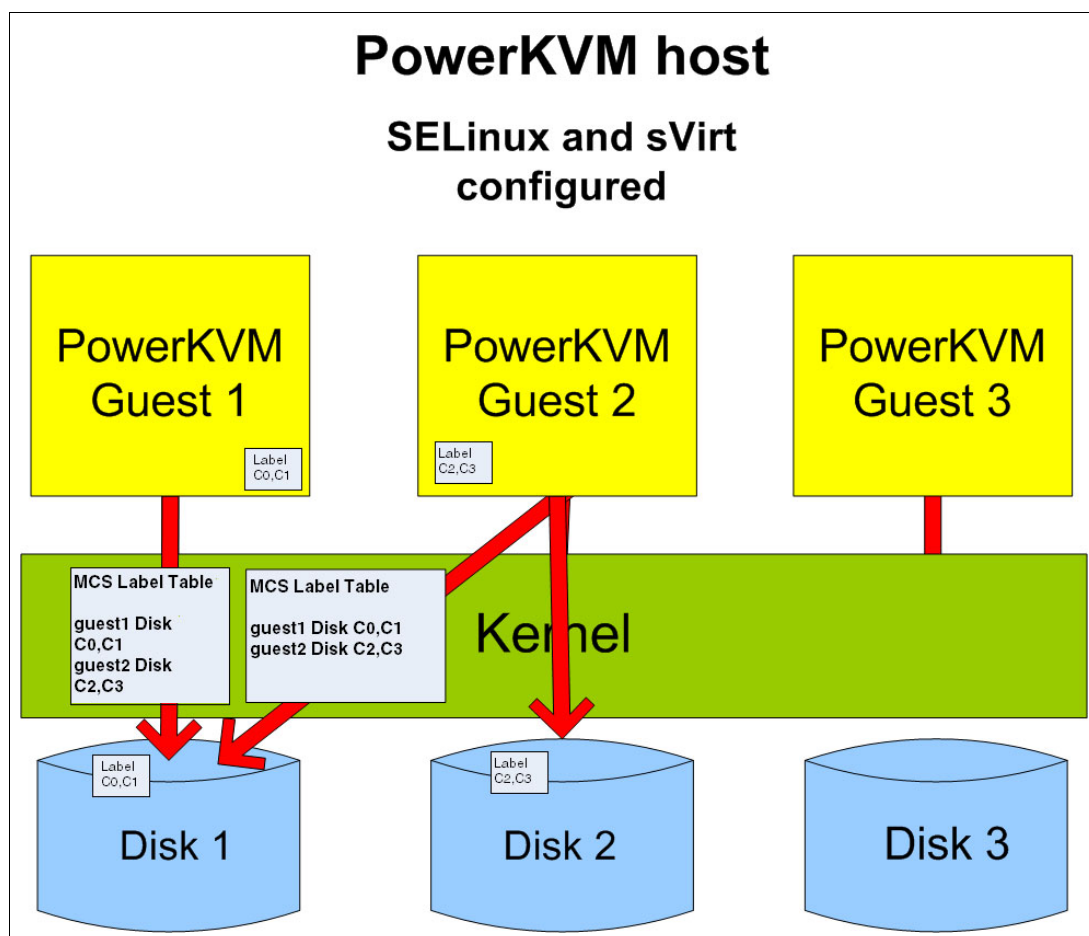


Figure 5-7 PowerKVM guest with access denied to another disk image

sVirt labeling

The sVirt services uses MCS labels to provide extra security and control over PowerKVM guests. MCS labels are applied automatically to resources on the system based on the currently running virtual machines (dynamic), but can also be manually specified by the system administrator (static).

Guests are born in the `svirt_t` domain, but with a unique category label (for example `c0,c1`). The disk resource for this guest, labeled with `system_u:object_r:svirt_image` has category `c0,c1` also. Therefore, this guest is able read and write to the `c0,c1` labeled image. Another guest has another category list, like `c2,c3`. As a result, this guest, even while running in the `svirt_t` domain, cannot access the image of the other guest, because the categories are different.

Five sVirt labels are available as listed in Table 5-4.

Table 5-4 The sVirt labels

Type	SELinux context	Description/Effect
Virtual Machine Process	system_u:system_r:svirt_t:MCS	MCS1 is a category field.
Virtual Machine Image	system_u:object_r:svirt_image_t:MCS	The svirt_t processes with the same MCS1 fields are able to read / write these image files and process.
Virtual Machine Shared Read/Write Content	system_u:object_r:svirt_image_t:s0	Processes are allowed to write in svirt_image_t:s0 files and process.
Virtual Machine Shared Read Only Content	system_u:object_r:svirt_content_t:s0	All svirt_t processes are able to read files and process with this label.
Virtual Machine Image	system_u:object_r:virt_content_t:s0	No svirt_t virtual processes are allowed to read files and process.

The default PowerKVM configuration allows each guest to run under its own dynamically configured domain and categories, isolating the PowerKVM guests from each other. Every time the system is rebooted, a different and unique MCS label is generated to confine each guest to its own domain. This process does not require any special configuration in the XML file.

Example 5-44 shows how QEMU processes and disk image files are dynamically labeled using sVirt.

Example 5-44 sVirt Dynamic labeling

```
[root@powerkvm7 /]# ps axZ | grep qemu

system_u:system_r:svirt_t:s0:c152,c669 27874 ? S1      6:41 /usr/bin/qemu-system-ppc64 -machine
accel=kvm -name PowerKVM_Guest_1
...

system_u:system_r:svirt_t:s0:c66,c800 39394 ? S1      0:05 /usr/bin/qemu-system-ppc64 -machine
accel=kvm -name PowerKVM_Guest_0
...

[root@powerkvm7 /]# ls -lZ /var/lib/libvirt/images/dbb9235e-fe8d-4cdc-ab47-c017cd5a0ed9-0.img
-rw----- . qemu qemu system_u:object_r:svirt_image_t:s0:c152,c669 ...

[root@powerkvm7 /]# ls -lZ /var/lib/libvirt/images/192f6731-c7c1-42b2-96ef-233037aeba35-0.img
-rw----- . qemu qemu system_u:object_r:svirt_image_t:s0:c66,c800 ...
```

PowerKVM_Guest_1 uses the categories c152,c669 and PowerKVM_Guest_2 uses c66,c800. The same values are assigned to the shared disk image files to obtain read and write access for them.

The system administrator can change the static labeling for a specific PowerKVM guest. The system administrator is responsible for setting the correct label on the disk image files. The guests always start with the assigned labels, and the sVirt system never modifies the labels.

Example 5-45 shows the XML configuration for static labeling.

Example 5-45 Adding XML configuration lines to static labeling

```
[root@powerkvm7 /]# virsh edit PowerKVM_Guest_0
...
<seclabel type='static' model='selinux'>
  <label>system_u:system_r:svirt_t:s0:c100,c200</label>
  <imagelabel>system_u:object_r:svirt_image_t:s0:c100,c200</imagelabel>
</seclabel>
```

After the XML file of the PowerKVM guest is configured, change the category values for the disk image file by running the following command.

```
chcon -t svirt_image_t -l s0:c100,c200
/var/lib/libvirt/images/192f6731-c7c1-42b2-96ef-233037aeba35-0.img
```

To validate the changes, do a guest restart:

```
virsh destroy PowerKVM_Guest_0
virsh start PowerKVM_Guest_0
```

The PowerKVM_Guest_0 is now running under a new static category. QEMU is running with c100,c200 values and disk image. Example 5-46 shows the new configuration.

Example 5-46 Verifying the new static labeling configuration

```
[root@powerkvm7 /]# ps axZ | grep qemu
sssystem_u:system_r:svirt_t:s0:c100,c200 42294 ? S1      0:38
/usr/bin/qemu-system-ppc64 -machine accel=kvm -name RHEL7_Guest2 ...

[root@powerkvm7 /]# ls -lZ
/var/lib/libvirt/images/192f6731-c7c1-42b2-96ef-233037aeba35-0.img
-rw----- . qemu qemu system_u:object_r:svirt_image_t:s0:c100,c200 ...
```

sVirt capability testing

To validate the sVirt capabilities and the protection for PowerKVM guests, create a new disk image file to be added as a new virtual disk for PowerKVM_Guest_0. Do the following steps:

1. Create the disk image file (disk2.img) with 5 GB:

```
# qemu-img create -f raw /var/lib/libvirt/images/disk2.img 5G
Formatting '/var/lib/libvirt/images/disk2.img', fmt=raw size=5368709120
```
2. Configure the new disk image in the XML configuration file for the PowerKVM_Guest_0 as shown in Example 5-47.

Example 5-47 Adding a new disk image for the PowerKVM guest

```
virsh edit PowerKVM_Guest_0
...
<disk type='block' device='disk'>
  <driver name='qemu' type='raw' />
  <source dev='/var/lib/libvirt/images/disk2.img' />
  <target dev='vdb' bus='virtio' />
</disk>
...
```

If you start the PowerKVM_Guest_0 at this time, the hypervisor returns a *permission denied* because the disk image file does not have the proper c100,c200 categories defined as configured earlier. Example 5-48 shows the lack of access for the disk image file.

Example 5-48 Permission denied for the new disk image file

```
root@powerkvm7 images]# virsh start PowerKVM_Guest_0
error: Failed to start domain PowerKVM_Guest_0
error: internal error: process exited while connecting to monitor:
qemu-system-ppc64: -drive
file=/var/lib/libvirt/images/disk2.img,if=none,id=drive-virtio-disk2,format=raw:
could not open disk image /var/lib/libvirt/images/disk2.img: Permission denied

[root@powerkvm7 images]# ls -lZ /var/lib/libvirt/images/disk2.img
-rw-r--r--. root root unconfined_u:object_r:virt_image_t:s0
/var/lib/libvirt/images/disk2.img
```

To obtain access to this new disk image, ensure that it is assigned the correct categories for PowerKVM_Guest_0, in this case, c100,c200. After that, you can start the guest as follows:

```
# chcon -t svirt_image_t -l s0:c100,c200 /var/lib/libvirt/images/disk2.img
[root@powerkvm7 images]# virsh start PowerKVM_Guest_0
Domain PowerKVM_Guest_0 started
```

Suggestion: When you enable the SELinux policy, the sVirt service automatically runs and dynamically creates and manages the labels. In most cases, sVirt dynamic labeling is recommended. However, you can disable dynamic labeling and create your own static labels. In this case, you are responsible for the uniqueness of the labels.

5.2.6 Audit

The ability to audit PowerKVM activities is a security capability that allows the tracking of changes and interactions as they happen in the PowerKVM environment. This tracking data can provide an audit trail for system security and can be valuable for forensic information in the event of an information breach. It also helps system administrators take additional security measures.

The audit integration is enabled by default on a PowerKVM host. It can be disabled and enabled by editing the `/etc/libvirt/libvirtd.conf` file. The configuration options are as follows:

- | | |
|----------------------|--|
| audit_level=0 | libvirt auditing is disabled regardless of host audit subsystem enablement. |
| audit_level=1 | libvirt auditing is enabled if the host audit subsystem is enabled, otherwise it is disabled. This is the default behavior. |
| audit_level=2 | libvirt auditing is enabled regardless of host audit subsystem enablement. If the host audit subsystem is disabled, then libvirtd refuses to complete startup and exits with an error. |

Example 5-49 shows how to configure audit_level 1.

Example 5-49 Editing /etc/libvirt/libvirtd.conf

```
[root@powerkvm7 grub2]# grep audit_ /etc/libvirt/libvirtd.conf
# Uncomment audit_level 1 and set to value 1
audit_level = 1

# If set to 1, then audit messages will also be sent
# via libvirt logging infrastructure. Defaults to 0
# Uncomment audit_logging line and set to value 1

audit_logging = 1
```

Restart the libvirtd daemon after the /etc/libvirt/libvirtd.conf file is configured:

```
systemctl restart libvirtd
```

Various situations can be audited. The following examples are auditable events:

- ▶ Changes of the TLS private keys.
- ▶ Changes of the libvirt configuration.
- ▶ Changes of the disk image files.
- ▶ Changes of the qemu and libvirt logs.
- ▶ Changes of the libvirtd daemon and execution time.
- ▶ Changes of the QEMU and execution time.

By using the /etc/audit/audit.rules file you can configure a list of arguments, one per line, to be passed to the auditctl tool. The rules are evaluated from top to bottom. For this reason, the order of appearance is important. Example 5-50 shows a typical audit rules file to monitor the PowerKVM environment.

Example 5-50 /etc/audit/audit.rules sample file

```
# First rule - delete all
-D

# Increase the buffers to survive stress events.
# Make this bigger for busy systems
-b 8192

#####
### Don't audit rules - explicit exclusions for more generic rules after
# Don't audit Qemu read/writes to necessary devices
-a exit,never -F path=/dev/kvm -F perm=rw -F subj_type=qemu_t
-a exit,never -F path=/dev/ksm -F perm=rw -F subj_type=qemu_t
-a exit,never -F path=/dev/ptmx -F perm=rw -F subj_type=qemu_t
-a exit,never -F dir=/dev/pts -F perm=rw -F subj_type=qemu_t

# Don't audit dnsmasq writing to libvirt network runtime data
-a exit,never -F dir=/var/run/libvirt/network -F perm=wa -F subj_type=dnsmasq_t

# Don't audit logrotate writing to logs
-a exit,never -F dir=/var/log/libvirt/ -F perm=wa -F subj_type=logrotate_t

# Don't audit initrc_t domain writing to temporary storage data
-a exit,never -F dir=/var/cache/libvirt/ -F perm=wa -F subj_type=initrc_t
```

```

#####
### Audit access attempts to TLS private keys
-a exit,always -F path=/etc/pki/libvirt/private/serverkey.pem -F subj_type!=virtd_t -k
virt_tls_privkey
-a exit,always -F path=/etc/pki/libvirt-vnc/server-key.pem -F subj_type!=qemu_t -k
virt_tls_privkey

#####
### Audit attempts at changing libvirt and Qemu certificates (both server and CA)
-a exit,always -F path=/etc/pki/CA/cacert.pem -F perm=wa -k virt_tls_cert
-a exit,always -F path=/etc/pki/libvirt/servercert.pem -F perm=wa -k virt_tls_cert
-a exit,always -F path=/etc/pki/libvirt-vnc/ca-cert.pem -F perm=wa -k virt_tls_cert
-a exit,always -F path=/etc/pki/libvirt-vnc/server-cert.pem -F perm=wa -k virt_tls_cert

#####
### Audit any changes to libvirt configuration
-a exit,always -F dir=/etc/libvirt/ -F perm=wa -k virt_libvirt_cfg
-a exit,always -F path=/etc/sysconfig/libvirtd -F perm=wa -k virt_libvirt_cfg
-a exit,always -F path=/etc/sasl2/libvirt.conf -F perm=wa -k virt_libvirt_cfg

#####
### Audit every attempt of qemu_t interaction with another domain, unless not
### explicitly excluded above
-a exit,always -F arch=b32 -S all -F perm=wax -F subj_type=qemu_t -F obj_type!=qemu_t -k
virt_qemu_crossdomain
-a exit,always -F arch=b64 -S all -F perm=wax -F subj_type=qemu_t -F obj_type!=qemu_t -k
virt_qemu_crossdomain

#####
### Audit changes to virtual images from outside qemu_t domain
-a exit,always -F dir=/var/lib/libvirt/images/ -F perm=wa -F subj_type!=qemu_t -k
virt_image_change
-a exit,always -F obj_type=virt_image_t -F perm=wa -F subj_type!=qemu_t -k virt_image_change

#####
### Audit changes to qemu/libvirt runtime data (exceptions above)
-a exit,always -F dir=/var/run/libvirt/ -F perm=wa -F subj_type!=virtd_t -k virt_runtime_change
-a exit,always -F dir=/var/lib/libvirt/ -F perm=wa -F subj_type!=virtd_t -k virt_runtime_change
-a exit,always -F dir=/var/cache/libvirt/ -F perm=wa -F subj_type!=qemu_t -k virt_runtime_change

#####
### Audit changes to qemu/libvirt logs (exceptions above)
-a exit,always -F dir=/var/log/libvirt/ -F perm=wa -F subj_type!=virtd_t -k virt_log_change

#####
### Audit every libvirtd execution
-a exit,always -F path=/usr/sbin/libvirtd -F perm=x -k virt_libvirtd_exec

#####
### Audit every libvirtd executable change
-a exit,always -F path=/usr/sbin/libvirtd -F perm=wa -k virt_libvirtd_change

#####
### Audit every Qemu execution
-a exit,always -F path=/usr/libexec/qemu-kvm -F perm=x -k virt_qemu_exec

```

```
#####
### Audit every Qemu executable change
-a exit,always -F path=/usr/libexec/qemu-kvm -F perm=wa -k virt_qemu_change

#####
### Record events that modify the System's Mandatory Access Controls
-w /etc/selinux/ -p wa -k MAC-policy

#####
### To keep /etc/audit/audit.rules configuration file immutable
-e 2
```

Restart both the auditd and libvirtd daemons after any configuration of the /etc/audit/audit.rules file:

```
systemctl restart auditd
systemctl restart libvirtd
```

To display all libvirt audit events from a specific PowerKVM guest, you can use the **auvirt** command as shown in Example 5-51.

Example 5-51 libvirt audit events of PowerKVM_Guest_0

```
[root@powerkvm7 audit]# auvirt --vm PowerKVM_Guest_0 --all-events
res  PowerKVM_Guest_0  root  Thu Aug 21 11:44  cgroup  deny  all
...
res  PowerKVM_Guest_0  root  Thu Aug 21 11:44  mem    start  4198400
res  PowerKVM_Guest_0  root  Thu Aug 21 11:44  vcpu   start  4
start PowerKVM_Guest_0  root  Thu Aug 21 11:44
```

The auditd daemon saves its logs in the /var/log/audit/audit.log file. Here you can find all audit tracking information pertaining to operational system activities, such as change of directory and file permission, access denied for files, and start and stop of processes.

The auditd daemon requires only modest disk space and does not affect system performance for most environments. The audit.log files are rotated every 6 MB in size and a new file is created. The older files are renamed to audit.log.1, audit.log.2, and so on.

Suggestion: Ensure that log files are backed up on a weekly schedule and that the audit files have permissions of 640 or more restrictive.

Example 5-52 shows how a permission change (**chmod**) on the libvirt path (`/etc/libvirt/`) is tracked. This action can also be identified by the key `virt_libvirt_cfg`, because it was configured in the `auditd` configuration file.

Example 5-52 Security audit testing

```
[root@powerkvm7 audit]# chmod 777 /etc/libvirt

[root@powerkvm7 audit]# tail -3 audit.log
type=SYSCALL msg=audit(1408661041.163:2295): arch=80000015 syscall=297 success=yes
exit=0 a0=ffffffffffffff9c a1=1001e120120 a2=1ff a3=0 items=1 ppid=25131 pid=62931
auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 ses=11 tty=pts0
comm="chmod" exe="/usr/bin/chmod"
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="virt_libvirt_cfg"
type=CWD msg=audit(1408661041.163:2295): cwd="/var/log/audit"
type=PATH msg=audit(1408661041.163:2295): item=0 name="/etc/libvirt" inode=395987
dev=fd:00 mode=040700 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:virt_etc_t:s0
nametype=NORMAL
```

To avoid any malicious change in the audit configuration file to mask an attack, ensure that the `/etc/audit/audit.conf` file cannot be changed after it is configured. A reboot is always required to change any audit rules after this step is implemented. Use the following command:

```
auditctl -e 2
```

You can create new audit entries to monitor other activities also. For more details about auditing and using the **ausearch** command to generate reports, see the following web page:

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/chap-system_auditing.html

5.2.7 PowerKVM guest image encryption

The PowerKVM guest image encryption is a mechanism that can protect information by converting it into unreadable code that cannot be deciphered by malicious attackers.

This section describes how to manage disk images for a PowerKVM guest and shows how to create a secure and reliable environment to avoid non-authorized access to disk image data.

The following topics are covered in this section:

- ▶ “Creating an encrypted logical volume” on page 130
- ▶ “Storing the encrypted logical volume” on page 132
- ▶ “Migrating existing PowerKVM guests to an encrypted device” on page 133

Creating an encrypted logical volume

This example creates an encrypted logical volume (LV) using a physical device called /dev/sdb1 and volume group (VG) called vg_images. Complete the following steps:

1. Create a physical disk, volume group, and logical volume. In Example 5-53, the logical volume is called PowerKVM_Guest_2-crypt and has 50 GB. It will be used as the back-end block device to be encrypted.

Example 5-53 Creating a physical device, volume group, and logical volume

```
# pvcreate /dev/sdb1
Physical volume "/dev/sdb1" successfully created

# vgcreate vg_images /dev/sdb1
Volume group "vg_images" successfully created

# lvcreate -L 50G -n PowerKVM_Guest_2-crypt vg_images
Logical volume "PowerKVM_Guest_2-crypt" created
```

2. Encrypt the new logical volume by using the **cryptsetup luksFormat** command as shown in Example 5-54.

Attention: Be sure that you are working with the correct new logical volume, otherwise the whole disk will be erased.

Example 5-54 Encrypting the new logical volume

```
# cryptsetup luksFormat /dev/vg_images/PowerKVM_Guest_2-crypt

WARNING!
=====
This will overwrite data on /dev/vg_images/PowerKVM_Guest_2-crypt irrevocably.

Are you sure? (Type uppercase yes): YES
Enter passphrase:
Verify passphrase:
```

3. Verify that the logical volume PowerKVM_Guest_2-crypt was formatted for encryption with success, as shown in Example 5-55.

Example 5-55 Verifying the status of the logical volume

```
# cryptsetup luksDump /dev/vg_images/PowerKVM_Guest_2-crypt
LUKS header information for /dev/vg_images/PowerKVM_Guest_2-crypt

Version:          1
Cipher name:      aes
Cipher mode:      xts-plain64
Hash spec:        sha1
Payload offset:   4096
MK bits:          256
MK digest:        49 a1 c7 0d 52 fc 8a 84 b8 85 56 f0 f7 8a 7d f7 a7 7e 60 ca
MK salt:          d1 5b ab d2 5e 48 87 63 d2 5e 60 d0 c6 cd bb f8
                  e8 d9 23 cb 41 f9 e1 5c 6f 3c 82 97 62 c6 c3 cd
MK iterations:    37625
UUID:             2f6ad240-26a9-4fee-aa4a-25dcda1a50b3

Key Slot 0: ENABLED
    Iterations:          148836
    Salt:                6a 24 c3 ac 94 f0 71 f1 56 ea 5e 30 c0 f2 91 cd
```



```

                                fe 50 0c fc a6 06 8f 6f 23 b9 67 9d 0d a1 d7 a7
Key material offset:          8
AF stripes:                   4000
Key Slot 1: DISABLED
Key Slot 2: DISABLED
Key Slot 3: DISABLED
Key Slot 4: DISABLED
Key Slot 5: DISABLED
Key Slot 6: DISABLED
Key Slot 7: DISABLED

```

4. Open the PowerKVM_Guest_2-crypt logical volume and create an encrypted device called powerkvm_guest_2. A passphrase is required to open (decrypt) this new device. Run the following command:

```
# cryptsetup luksOpen /dev/vg_images/PowerKVM_Guest_2-crypt powerkvm_guest_2
Enter passphrase for /dev/vg_images/PowerKVM_Guest_2-crypt:
```

5. Verify the status of the newly created powerkvm_guest_2 logical volume. The ACTIVE status (Example 5-56) indicates that the disk is opened and decrypted, and that it can be used.

Example 5-56 Verifying the powerkvm_guest_2 logical volume

```
# dmsetup info powerkvm_guest_2
Name:                powerkvm_guest_2
State:             ACTIVE
Read Ahead:          256
Tables present:      LIVE
Open count:           0
Event number:         0
Major, minor:         253, 5
Number of targets:    1
UUID: CRYPT-LUKS1-2f6ad24026a94feaa4a25dcda1a50b3-powerkvm_guest_2
```

6. Create an ext4 file system and mount the unencrypted logical volume to a specific path as shown in Example 5-57.

Example 5-57 Creating and mounting the new filesystem

```
# mkfs.ext4 /dev/mapper/powerkvm_guest_2
mke2fs 1.42.7 (21-Jan-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
3276800 inodes, 13106688 blocks
655334 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
400 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424
Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
# mount /dev/mapper/powerkvm_guest_2 /images/PowerKVM_Guest_2

# df -k /images/PowerKVM_Guest_2
Filesystem              1K-blocks  Used Available Use% Mounted on
/dev/mapper/powerkvm_guest_2  51473020  53064   48782236   1% /images/PowerKVM_Guest_2
```

7. The new /images/PowerKVM_Guest_2 can be mapped as a specific storage pool for the virtual machine images as demonstrated in Example 5-58.

Example 5-58 Creating a new PowerKVM guest

```
virt-install --name PowerKVM_Guest_2 \
--ram 2048 --vcpus=2 \
--disk path=/images/PowerKVM_Guest_2/guest_2.img,size=49,format=qcow2 \
--network network:Prod --accelerate --vnc \
--os-type=linux --os-variant=rhel7 \
--location=/var/lib/libvirt/RHEL-7.0-20140507.0-Server-ppc64-dvd.iso

Starting install...
Retrieving file .treeinfo...          | 2.8 kB  00:00:00 !!!
Retrieving file vmlinuz...            | 38 MB   00:00:00 !!!
Retrieving file initrd.img...         | 63 MB   00:00:00 !!!
Allocating 'powerkvm_guest_2.img'    | 49 GB   00:00:00
Creating domain...                   | 0 B     00:00:00
WARNING Unable to connect to graphical console: virt-viewer not installed. Please
install the 'virt-viewer' package.
Domain installation still in progress. You can reconnect to
the console to complete the installation process.
```

Expected warning: The warning message shown in Example 5-58 is expected because PowerKVM does not have the virt-viewer tool installed. You must use the Kimchi web interface to get a console of the virtual machine and continue the installation process.

Storing the encrypted logical volume

To store disk image files in a secure mode when a virtual machine is offline and protect your data from attackers, you must encrypt the logical volume. That way no one will have access to internal files used on the PowerKVM guest. Complete the following steps:

1. Shut down the PowerKVM guest and unmount the file system where the disk image file is stored, as shown in Example 5-59.

Example 5-59 Shutting down the PowerKVM guest and unmounting the file system

```
# virsh destroy PowerKVM_Guest_2
Domain PowerKVM_Guest_2 destroyed

# umount /images/PowerKVM_Guest_2
```

2. Close the encrypted device by using the following command.

```
cryptsetup luksClose powerkvm_guest_2
```

The encrypted device and logical volumes are unavailable for use now as shown in Example 5-60.

Example 5-60 Verifying encrypted device and logical volumes are unavailable

```
# mount /dev/mapper/powerkvm_guest_2 /images/PowerKVM_Guest_2
mount: special device /dev/mapper/powerkvm_guest_2 does not exist

# mount /dev/vg_images/PowerKVM_Guest_2-crypt /images/PowerKVM_Guest_2
mount: unknown filesystem type 'crypto_LUKS'
```

Migrating existing PowerKVM guests to an encrypted device

If you already have PowerKVM guests running and want to migrate them to an encrypted structure, the suggestion is to perform a *cold migration*, shutting down virtual machines prior to the migration steps. Complete the following steps:

1. Shut down the PowerKVM guest:
`virsh destroy PowerKVM_Guest_0`
2. Find the current location of the disk image for the PowerKVM guest to be migrated. Verify guest configuration by inspecting the `<disk>` tags under the `<device>` section, as shown in Example 5-61. In this example the primary disk is placed at the `/var/lib/libvirt/images` location.

Example 5-61 Verifying PowerKVM guest XML configuration

```
# virsh dumpxml PowerKVM_Guest_0
...
<devices>
  <emulator>/usr/bin/qemu-kvm</emulator>
  <disk type='file' device='disk'>
    <driver name='qemu' type='raw' cache='none' />
    <source
      file='/var/lib/libvirt/images/b684030c-eb30-45bd-a891-f4cec1d2774b-0.img' />
    <target dev='sda' bus='scsi' />
    <address type='drive' controller='0' bus='0' target='0' unit='0' />
  </disk>
  ...
```

3. Verify the disk image size and format of the virtual disk, as shown in Example 5-62.

Example 5-62 Verifying the disk image

```
# qemu-img info
/var/lib/libvirt/images/b684030c-eb30-45bd-a891-f4cec1d2774b-0.img
...
file format: raw
virtual size: 50G (53687091200 bytes)
disk size: 50G
```

4. Create a new encrypted logical volume as demonstrated in “Creating an encrypted logical volume” on page 130, then move the disk image file to the mount point of the encrypted target device by using the `mv` utility:

```
# mv /var/lib/libvirt/images/b684030c-eb30-45bd-a891-f4cec1d2774b-0.img
/images/PowerKVM_Guest_0/
```

5. Edit the PowerKVM guest configuration to point to its new storage location. Note the changes to the disk type from file to *block*, and the disk source to *dev* instead of file, as shown in Example 5-63:

Example 5-63 Editing the disk type and source file

```
# virsh dumpxml PowerKVM_Guest_2
...
<emulator>/usr/bin/qemu-kvm</emulator>
  <disk type='block' device='disk'>
    <driver name='qemu' type='qcow2' />
    <source file='/images/PowerKVM_Guest_2/powerkvm_guest_2.img' />
    <target dev='sda' bus='scsi' />
    <address type='drive' controller='0' bus='0' target='0' unit='0' />
  </disk>
...
```

6. Start the PowerKVM guest:

```
virsh start PowerKVM_Guest_2
```

Suggestion: When you do not have a virtual machine working and available, use encrypted logical volumes to make sure that all data is protected against malicious attacks.

5.2.8 Guest live migration

The PowerKVM guest live migration helps you to migrate a virtual machine from one PowerKVM host to another. This is commonly used to avoid business impact because of scheduled maintenance or an unexpected problem. This capability can also be used as a scalability feature to move around workloads between hosts.

Before you do the live migration activity, examine four important aspects:

- ▶ The guest disk image must be shared on networked storage using one of the following protocols:
 - Fibre Channel
 - NFS
 - iSCSI
- ▶ The PowerKVM hosts must be at the same software level.
- ▶ The PowerKVM hosts must have identical network configurations.
- ▶ The PowerKVM hosts must use the same mount points used by the hypervisor.

The guest live migration must be performed using an SSH tunnel to encrypt the traffic between both PowerKVM hosts.

The following steps demonstrate how to configure the hosts and migrate a virtual machine from the powerkvm7 (source) to the powerkvm8 (destination) server:

1. Be sure that you deployed the same software levels, and also network and storage pool configurations, on both PowerKVM hosts.
2. Be sure that you have a disk image file stored in a shared location. In this example, NFS is used.

3. Create SSH private and public keys on the source server as shown in Example 5-64.

Example 5-64 Creating SSH keys in the source server

```
[root@powerkvm7 /]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
ad:63:6b:b8:7f:f1:58:d1:ce:54:ca:6f:47:6e:58:8a root@powerkvm7.itso.ibm.com
The key's randomart image is:
+--[ RSA 2048 ]-----+
|
|          .
|        o o
|       . . = o
|      S . * B
|     .. E = *
|    .+ = o.
|   ...oo .
|  .+o.
|
+-----+
```

4. Rename the public key to `authorized_keys` and copy it to the destination server by using the following commands.

```
[root@powerkvm7 /]# mv /root/.ssh/id_rsa.pub /root/.ssh/authorized_keys
[root@powerkvm7 /]# scp /root/.ssh/id_rsa.pub root@powerkvm8:/root/.ssh
```

5. Before validating this setup, run a monitoring and performance tool, named **top**, in the background by using the following commands:

```
[root@powerkvmguest ~]# top &
[1] 7613
[root@powerkvmguest ~]# ps auxww | grep top
root      7613  0.0  0.1 112192  4224 pts/0    T   20:31   0:00 top
```

6. Migrate the virtual machine `PowerKVM_Guest_2` from the `powerkvm7` to the `powerkvm8` server by using a secure SSH tunnel as shown in Example 5-65.

Example 5-65 Migrating virtual machine to powerkvm8 server

```
[root@powerkvm7 /]# virsh list --all
 Id      Name                               State
-----
 51      PowerKVM_Guest_2                  running
[root@powerkvm7 /]#
[root@powerkvm7 /]# virsh migrate --live --persistent --undefinesource
--verbose --p2p --tunnelled PowerKVM_Guest_2 qemu+ssh://root@powerkvm8/system
Migration: [100 %]
[root@powerkvm7 /]
```

7. Validate the migration process by verifying that the virtual machine is running on the destination server and that the top process is running as shown in Example 5-66.

Example 5-66 Validating the migration

```
[root@powerkvm8 /]# virsh list --all
  Id      Name                               State
-----
   51     PowerKVM_Guest_2 running
[root@powerkvm8 /]#
[root@powerkvmguest ~]# ps auxww | grep top
root      7613  0.0  0.1 112192  4224 pts/0    T   20:31   0:00 top
```

Suggestion: If a PowerKVM host uses multiple network interfaces or if the network switch supports tagged VLANs, then a good approach is to separate guest network traffic from migration traffic or use only a management network. This approach helps avoid a malicious attack to obtain guest information during a live migration.

5.3 Conclusion

This chapter shows how to set up IBM PowerKVM functional components for proper security measures. The most important security capabilities are discussed and some suggestions are given to ensure a safe and reliable environment.

As a summary, consider the following individual steps:

- ▶ Configure secure remote management using strong encryption.
- ▶ Configure VLAN tagging to separate the internal networks.
- ▶ Enable network filter configuration to avoid spoofing attacks.
- ▶ Enable sVirt utilization to isolate PowerKVM guests.
- ▶ Configure restricted permissions for the disk images.
- ▶ Encrypt your disk images.
- ▶ Enable audit mode.

IBM PowerKVM also works with IBM PowerVC to deliver comprehensive solutions for server provisioning and to help simplify configuration and change management, and to reduce operational costs. For more information about PowerVC, see Chapter 6, “IBM PowerVC security” on page 137.



IBM PowerVC security

This chapter introduces the IBM advanced virtualization management solution for IBM Power Systems, PowerVC built on OpenStack, from a security perspective, and discusses how to configure and implement the PowerVC environment to ensure optimal security. The guidelines and tips are based on the PowerVC 1.3 release, which can be installed on SELinux.

The following topics are covered in this chapter:

- ▶ 6.1, “Introduction to PowerVC and security topics” on page 138
- ▶ 6.2, “Identity management” on page 153
- ▶ 6.3, “API security” on page 158
- ▶ 6.4, “Audit” on page 162
- ▶ 6.5, “Security options using powervc-config command” on page 165
- ▶ 6.6, “Patch management” on page 167
- ▶ 6.7, “Conclusion” on page 168

6.1 Introduction to PowerVC and security topics

PowerVC is the advanced virtualization management offering, built on OpenStack, that delivers advanced virtualization management for IBM AIX, IBM i, and Linux environments on IBM Power Systems. As shown in Figure 6-1, PowerVC is based on the major components of OpenStack and has been modified and extended to support IBM Power Systems. Also, the dashboard of OpenStack, Horizon, is replaced with the PowerVC Virtualization Management Console.

From a security perspective, implementing and managing PowerVC securely is similar to OpenStack. The OpenStack Security Guide¹ describes many of the aspects of OpenStack security.

Note: PowerVC automates the installation and configuration with security enforced, by default, whereas OpenStack must be manually secured.

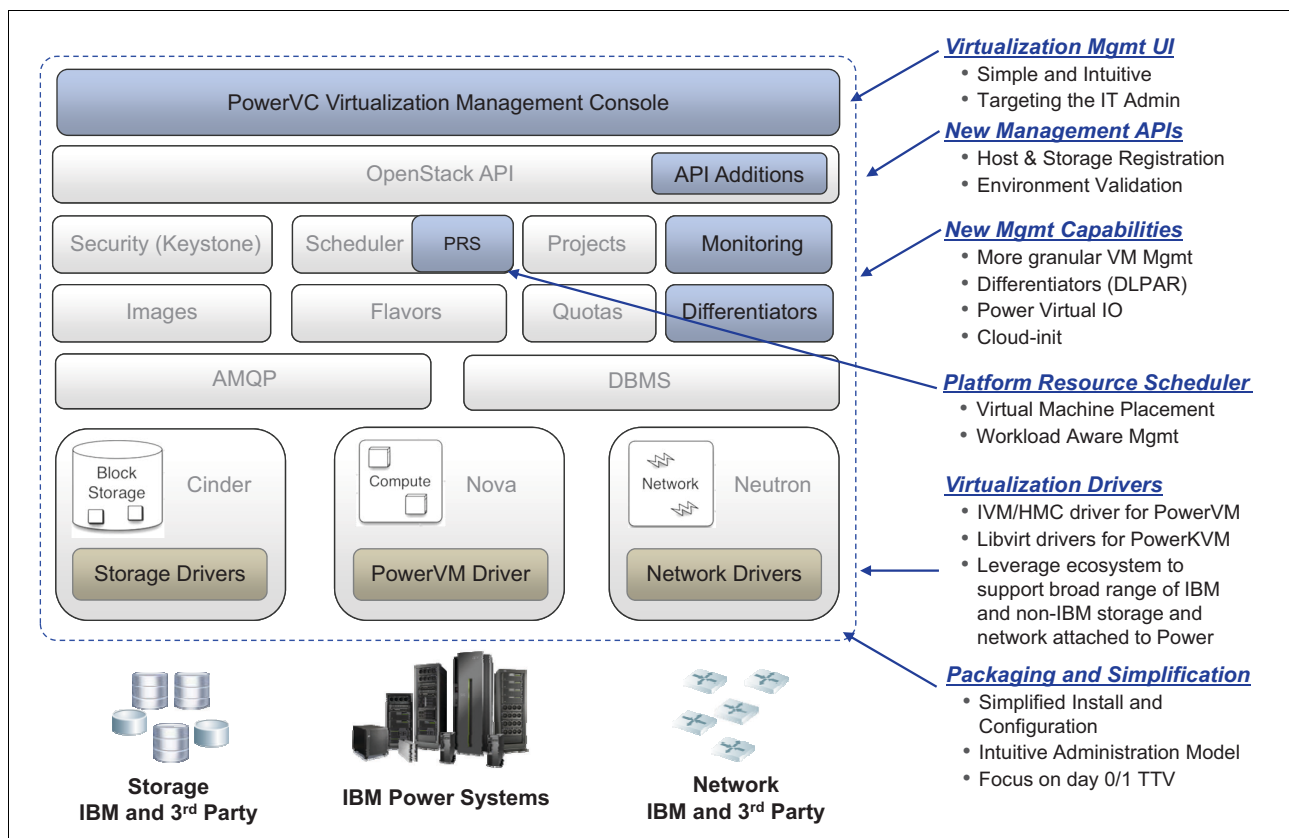


Figure 6-1 PowerVC component architecture

¹ OpenStack Security Guide provides best practices and conceptual information about securing an OpenStack Cloud: <http://docs.openstack.org/sec/>

Because OpenStack leaves security to the implementor or deployer, PowerVC introduces additional security features that enhance security of OpenStack. For example, PowerVC enhances security in the following components:

- ▶ PowerVC console user management
- ▶ Secure communication channels
- ▶ Audit management
- ▶ Patch management

This chapter, describes the security features and considerations in implementing and managing a PowerVC environment. The following PowerVC topics are covered in this section:

- ▶ 6.1.1, “PowerVC architecture overview” on page 139
- ▶ 6.1.2, “Security enhancement features” on page 143
- ▶ 6.1.3, “Secure communications” on page 151

6.1.1 PowerVC architecture overview

Figure 6-2 depicts the PowerVC architecture in a PowerVM environment and its interaction with resources, cloud management solutions, and API users.

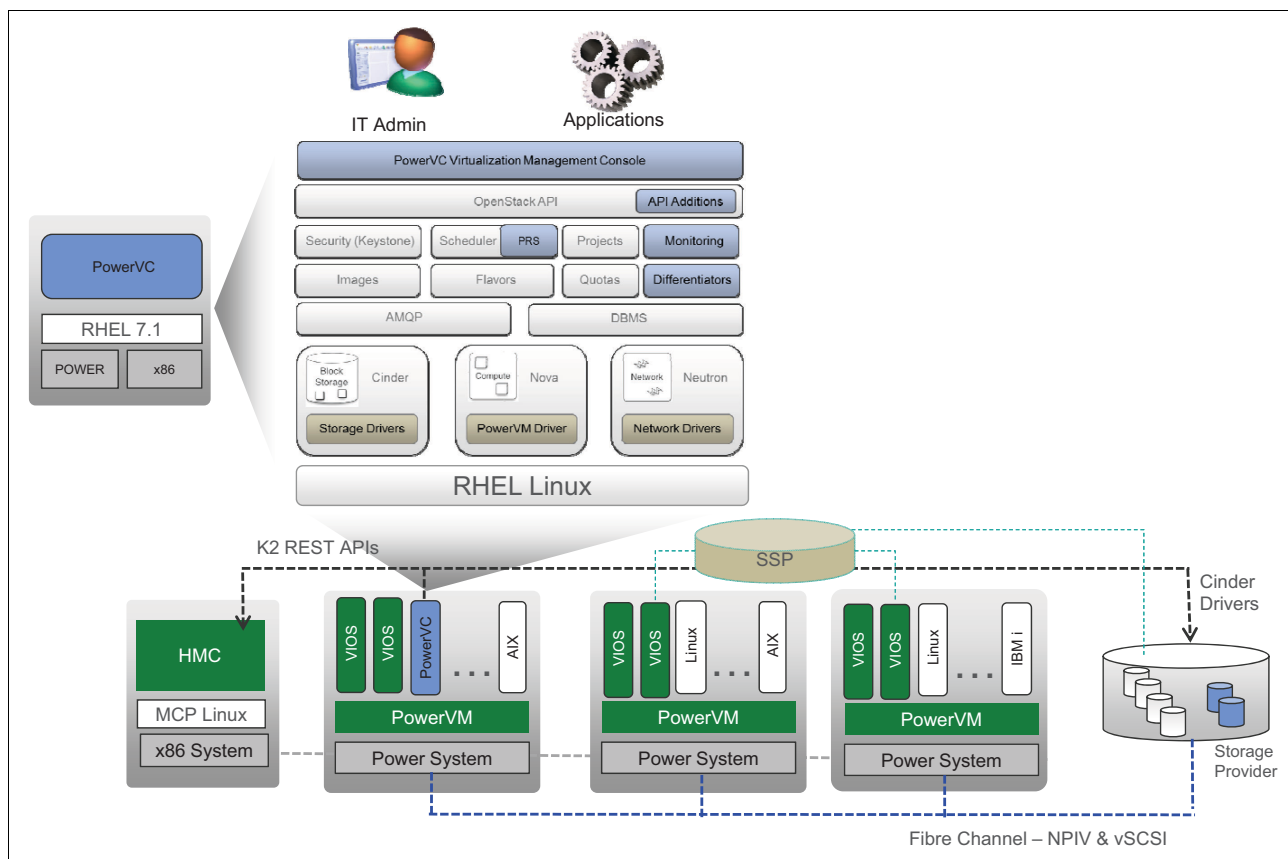


Figure 6-2 Architecture of PowerVC managing PowerVM

The diagram illustrates the OpenStack architecture on PowerVC. At the top, **IT Admin** and **Applications** interact with the **PowerVC Virtualization Management Console**. The console provides an **OpenStack API** (with **API Additions**) and manages various services: **Security (Keystone)**, **Scheduler** (with **PRS**), **Projects**, **Monitoring**, **Images**, **Flavors**, **Quotas**, and **Differentiators**. It also handles **AMQP** and **DBMS**. The core services are **Cinder** (with **Storage Drivers**), **Nova** (with **PowerVM Driver**), and **Neutron** (with **Network Drivers**). These services run on **RHEL Linux**. The **PowerVC** layer is based on **RHEL 7.1** and supports **POWER** and **x86** architectures. The architecture is distributed across multiple **POWER8** nodes, each running **RHEL 6**, **SLES 11**, **RHEL 7**, and **Ubuntu LE***. Each node includes a **KVM Kernel**, **OPAL FW**, and **POWER8** hardware. A **Network File Server** is connected to the nodes. **Cinder Drivers** are connected to a **Storage Provider** via **iSCSI**.

Figure 6-3 Architecture of PowerVC managing PowerKVM

Figure 6-4 depicts the PowerVC architecture in a PowerVM environment using NovaLink.

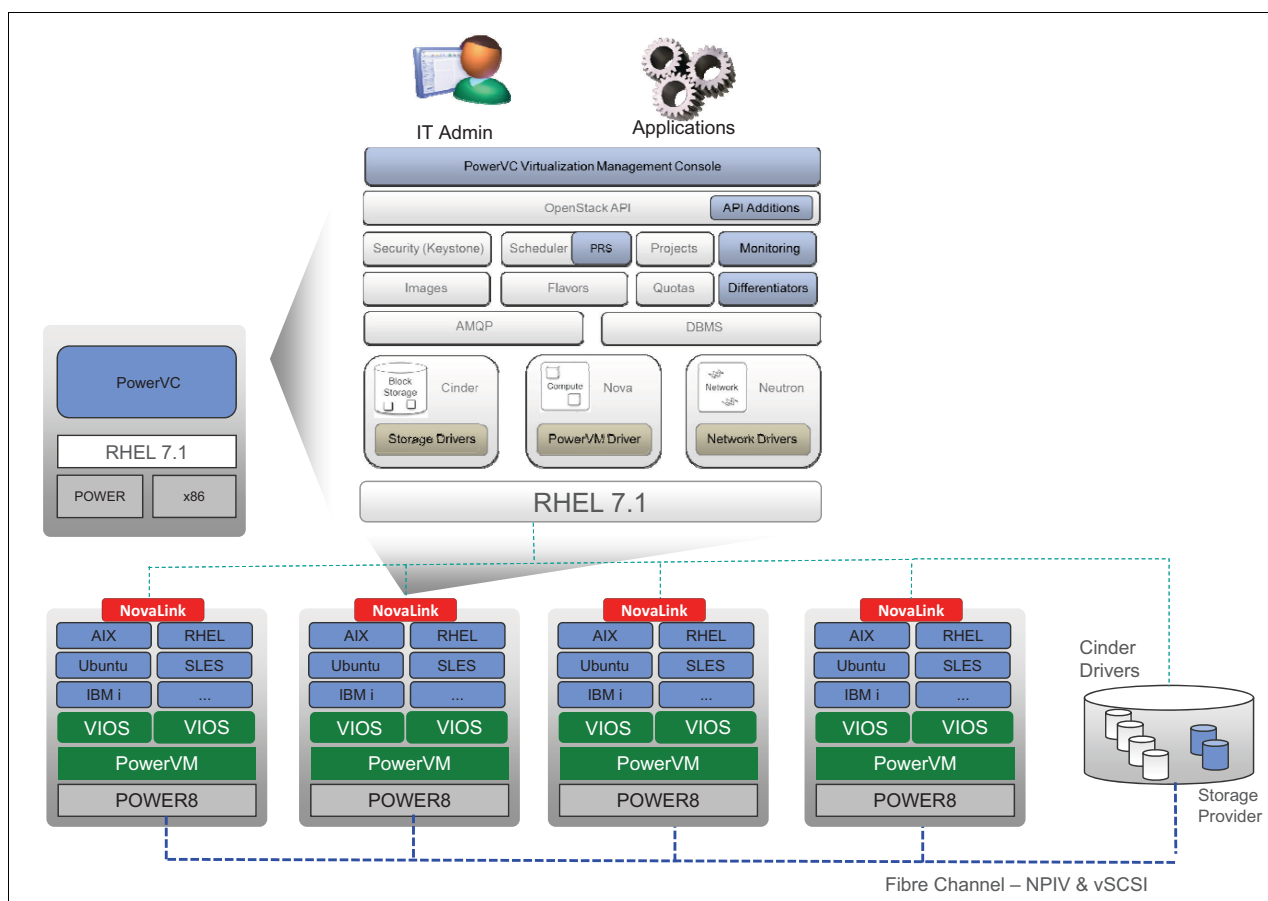


Figure 6-4 Architecture of PowerVC managing PowerVM using NovaLink

PowerVC implements a subset of OpenStack components that are necessary to provide infrastructure as a service (IaaS) cloud services for IBM Power Systems. The current PowerVC deployment architecture only allows for the deployment of all components into a single management server, which makes it simple to secure PowerVC.

The PowerVC components and their relationship with OpenStack are described next. By being built on OpenStack, Power Systems are opened up to cloud solutions through PowerVC.

PowerVC web user interface

PowerVC has its own web UI replacing OpenStack Horizon (the web dashboard component in OpenStack) providing advanced virtualization management features for IBM Power Systems.

The access to this web UI is protected by a user ID and password. Although the UI is accessible by HTTP protocol, it is designed to automatically redirect HTTP requests to HTTPS to ensure a secure channel between the user and the PowerVC web UI. Users can access only the functions based on their granted group role. For more information about groups and roles, see 6.2.2, “PowerVC users, groups, roles, and policies” on page 155.

Keystone

OpenStack Keystone provides the identity service to OpenStack for authentication and authorization. Authentication tokens are distributed to authorized users upon successful login. These tokens are used for subsequent service requests.

Keystone also provides catalog services. From a security perspective, Keystone operates like a gate keeper to the OpenStack environment. Keystone stores user information in its own database or in the local operating system, or delegates authentication to LDAP directory services.

PowerVC uses the user management capability of the operating system on the management server or an LDAP² directory server for authentication. PowerVC does not provide the option to store end user information in a local database, which is a method used only by OpenStack.

PowerVC implements the Identity API v3 for enhanced authentication and authorization, which allows Pluggable Authentication Module and multiple groups, and supports security policies.

PowerVC defines three groups and roles that have separate capabilities:

- ▶ admin
- ▶ deployer
- ▶ viewer

More details about these groups are described in 6.2.2, “PowerVC users, groups, roles, and policies” on page 155.

OpenStack command-line tools are not compatible with the PowerVC role-based security. Instead, a PowerVC version of the Keystone CLI, **powervc-keystone** exists. For other API access methods, use the cURL or REST API clients.

PowerVC uses two built-in projects (or tenants) and one domain, which cannot be customized.

API services

PowerVC provides the Northbound API services as a RESTful HTTP API so that cloud solutions can integrate with IBM Power Systems. PowerVC includes a subset of OpenStack APIs and its own APIs to provide IBM Power System specific services.

As with OpenStack APIs, users must first be authenticated with a user ID and password and then receive the authentication token, for which the expiration is customized (default is six hours.) Users can send API requests along with the given authentication token. API access is authorized based on the policy per the assigned role.

By default, public access to API communication is secured using the HTTPS protocol, and does not allow HTTP access.

UUID token: The PowerVC implementation uses a Universally Unique Identifier (UUID) token instead of a PKI based token. This cannot be changed.

Audit

To implement enhanced security and to be compliant with certain security standards, many organizations are required to implement and continuously monitor an audit subsystem. OpenStack supports a security audit feature by Ceilometer. To provide interoperability with

² PowerVC LDAP support includes OpenLDAP and Microsoft Active Directory.

security management systems, such as Security Information and Event Management (SIEM) systems, Ceilometer supports the Distributed Management Task Force (DMTF³) Cloud Auditing Data Federation (CADF⁴) standard.

For more information, see the blueprint of the OpenStack audit feature:

<http://wiki.openstack.org/wiki/Ceilometer/blueprints/support-standard-audit-formats>

PowerVC adopts this audit feature and provides audit capabilities based on request and response events. The audit data stored in Ceilometer's audit repository can be retrieved using the PowerVC command **powervc-audit-export**. You can enable the audit feature using the PowerVC command **powervc-config general audit** for compliance reasons and because of security visibility, which are critical from a practical security management perspective.

Time is important: Synchronizing the clock of the PowerVC management server and the managed servers is important so that audit log information from separate servers can be correlated in the correct chronological order. Use a public system or your internal NTP system when possible.

PowerVC integration with cloud solutions

As mentioned in 1.3.3, "Advanced virtualization management" on page 8, PowerVC provides advanced virtualization management. Being built on OpenStack, it opens the Power System cloud infrastructure to a broad range of cloud management solutions.

Separate network for virtualization and cloud management

Another architectural point to consider when you implement PowerVC is the separation of the virtualization management network from the network used for user applications. Within this separated network, PowerVC is deployed along with other management entities such as HMC, KVM host, SAN switch, and so on. API services should also be segregated to the management network.

Operating system of the management host

PowerVC 1.3 runs on Red Hat Enterprise Linux Server 7.1 and is supported on POWER and on x86. A suggestion is to be sure that you properly harden and secure that Red Hat Enterprise Linux Server before PowerVC is committed into production. For security and stability reasons, another suggestion is *not* to install software other than PowerVC to your management server for security and stability reasons.

6.1.2 Security enhancement features

As mentioned previously, PowerVC based on OpenStack adds security features to improve OpenStack security by default, which is documented in the design principle of the IBM Cloud Security Reference Architecture. This section provides an overview of these added features:

- ▶ "Encrypted user passwords, tokens, and persisted strings" on page 144
- ▶ "Firewall configured during IBM PowerVC installation" on page 144
- ▶ "Secure communication channels for PowerVC API endpoints" on page 149
- ▶ "Audit log and audit management tool" on page 150
- ▶ "Certification verification enforcement" on page 150

³ For information about the DMTF: <http://www.dmtf.org>

⁴ For information about the CADF: <http://www.dmtf.org/standards/cadf>

Encrypted user passwords, tokens, and persisted strings

PowerVC securely generates random passwords for each service user account during the installation phase. These passwords are stored in an encrypted format using 128-bit AES-CTR encryption to prevent them from being exposed accidentally.

Example 6-1 shows the encrypted neutron admin password defined in the nova configuration file.

Example 6-1 Encrypted neutron admin password defined in the nova.conf file

```
...
neutron_admin_password =
aes-ctr:NzMwODkzNTUzMTIzODgyNjY5MzoHmWY2qBHdzZPs6178W4q3Ed/9HQ==
...
```

In Example 6-2, SQL connection information (IP, user name, password, database name), neutron admin password, and admin password are stored in encrypted format.

Example 6-2 Encrypted password and persisted strings

```
...
sql_connection =
aes-ctr:MTU1Nzg0Mjc3MzQwNzUxMzc1ODpAjVD1QApl/s0tZt0szsgIUuUb/3kk1X6/01VfgkgS0/VStz
LZr5XzC3kbcZVd4MpRN450QJcFBVdJ
...
neutron_admin_password =
aes-ctr:NzMwODkzNTUzMTIzODgyNjY5MzoHmWY2qBHdzZPs6178W4q3Ed/9HQ==
...
admin_password = aes-ctr:MTUzMjc5Nzk4MzE2MTYzNDA4MzrMS3UkmDV/mjK8v8di1PWVfUh/5A==
...
```

Note: Although users can generate new encrypted passwords for service user accounts and replace old passwords with new passwords, it is not recommended because the new passwords must be updated to the operating system user registry, Keystone database, and OpenStack service configuration files.

Firewall configured during IBM PowerVC installation

IBM PowerVC is running on a Red Hat Linux server and provides a web-based user interface and OpenStack based REST API services. Applying proper access control to those services is important.

Table 6-1 lists network ports that are used by PowerVC for inbound and outbound traffic.

Table 6-1 Inbound and outbound ports used by PowerVC for external communication

Traffic direction	Port (TCP)	Usage	Protocol
Inbound	1 ^a	ping	ICMP
Inbound	80 ^b	Apache HTTPD Web Server	HTTP
Inbound	443	Apache HTTPD Web Server	HTTPS
Inbound	5000	Keystone	HTTPS
Inbound	5470	bumblebee	HTTPS

Traffic direction	Port (TCP)	Usage	Protocol
Inbound	5671	rabbitmq	AMQPS (AMQP over SSL/TLS)
Inbound	8428	validator	HTTPS
Inbound	8774	nova	HTTPS
Inbound	8777	ceilometer	HTTPS
Inbound	9000	cinder	HTTPS
Inbound	9292	glance	HTTPS
Inbound	9696	neutron	HTTPS
Inbound	35357	Keystone	HTTPS
Outbound	22	Brocade and Cisco Fibre Channel Switches, the IBM Storwize® family, and Novalink	SSH
Outbound	389	LDAP client	LDAP
Outbound	443	EMC VNX	SSH
Outbound	636	LDAP client	LDAPS
Outbound	5989	EMC VMAX	HTTPS
Outbound	7778	IBM XIV®	SSL
Outbound	8452	IBM System Storage DS8000	HTTPS
Outbound	12443 ^c	HMC	HTTPS

a. Only necessary with NovaLink managed hosts.

b. In the base configuration, all port 80/TCP HTTP web service requests are redirected to port 443/TCP HTTPS web service to ensure all the network traffic to be protected using strong cryptographic protocols. Port 80/TCP can be completely blocked if you want.

c. Only necessary with HMC managed hosts.

Table 6-2 lists network ports used internally within PowerVC.

Table 6-2 Ports used internally within the PowerVC management host

Port	Usage
4369	epmd
4952	ceilometer-collector
7869	lim
7870	vemkd
7871	pem
7872	egosc
9191	glance-registry
25672	rabbitmq-dist
50110	IBM DB2®

Table 6-3 shows network ports used on NovaLink managed hosts.

Table 6-3 Ports used on NovaLink managed hosts

Traffic direction	Port	Usage	Protocol
Inbound	22	Secure shell	SSH
Outbound	5000	Keystone	HTTPS
Outbound	5671	rabbitmq	AMQPS
Outbound	9292	glance	HTTPS
Outbound	9696	neutron	HTTPS

For PowerVC to successfully register a PowerKVM host, the PowerKVM host firewall must allow inbound traffic for port 22. The PowerKVM host firewall is automatically configured to allow libvirt-tls traffic during registration and also during migration, if necessary. You can configure the firewall before or after your register the host.

All the ports that are listed in Table 6-4 are required.

Table 6-4 Ports used on PowerKVM managed hosts

Traffic direction	Port	Usage	Protocol
Inbound	22	Secure shell	SSH
Inbound	16514	libvirt management	libvirt-tls
Outbound	111	NFS	NFS
Outbound	2049	NFS	NFS
Outbound	3260	SVC iSCSI	iSCSI
Outbound	5000	Keystone	HTTPS
Outbound	5671	rabbitmq	AMQPS
Outbound	9292	glance	HTTPS
Outbound	9696	neutron	HTTPS
Outbound	16514	libvirt management	libvirt-tls

By default, IBM PowerVC 1.3 automatically sets up the iptables firewall rules that allow only the necessary inbound and outbound traffic. This automatic iptables configuration can be disabled by using the **-c nofirewall** flag with the **install** script (Example 6-3). You can customize iptables anytime after PowerVC is installed.

Example 6-3 Installing PowerVC without the firewall enabled

```
# ./install -c nofirewall
```

Example 6-4 depicts the iptables firewall rules after PowerVC installation. The inbound access control is implemented as a *deny all* policy while outbound connection is not strictly restricted, although outbound rules are added.

Example 6-4 Default iptables rules defined by PowerVC

```
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- anywhere anywhere state NEW,ESTABLISHED tcp dpt:12443
ACCEPT tcp -- anywhere anywhere state NEW,ESTABLISHED tcp dpt:openstack-id
ACCEPT tcp -- anywhere anywhere state NEW,ESTABLISHED tcp dpt:amqps
ACCEPT tcp -- anywhere anywhere state NEW,ESTABLISHED tcp dpt:5470
ACCEPT tcp -- anywhere anywhere state NEW,ESTABLISHED tcp dpt:https
ACCEPT tcp -- anywhere anywhere state NEW,ESTABLISHED tcp dpt:http
ACCEPT tcp -- anywhere anywhere state NEW,ESTABLISHED tcp dpt:ssh
ACCEPT tcp -- anywhere anywhere state NEW,ESTABLISHED tcp dpt:8428
ACCEPT tcp -- anywhere anywhere state NEW,ESTABLISHED tcp dpt:8774
ACCEPT tcp -- anywhere anywhere state NEW,ESTABLISHED tcp dpt:armtechdaemon
ACCEPT tcp -- anywhere anywhere state NEW,ESTABLISHED tcp dpt:cslistener
ACCEPT tcp -- anywhere anywhere state NEW,ESTABLISHED tcp dpt:9696
ACCEPT tcp -- anywhere anywhere state NEW,ESTABLISHED tcp dpt:complex-main
ACCEPT tcp -- anywhere anywhere state NEW,ESTABLISHED tcp dpt:8777
ACCEPT udp -- anywhere anywhere udp dpt:domain
ACCEPT tcp -- anywhere anywhere tcp dpt:domain
ACCEPT udp -- anywhere anywhere udp dpt:bootps
ACCEPT tcp -- anywhere anywhere tcp dpt:bootps
ACCEPT all -- anywhere anywhere state RELATED,ESTABLISHED
ACCEPT icmp -- anywhere anywhere
ACCEPT all -- anywhere anywhere
ACCEPT tcp -- anywhere anywhere state NEW tcp dpt:ssh
ACCEPT tcp -- anywhere anywhere state NEW tcp dpt:5901
REJECT all -- anywhere anywhere reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target prot opt source destination
ACCEPT all -- anywhere 192.168.122.0/24 ctstate RELATED,ESTABLISHED
ACCEPT all -- 192.168.122.0/24 anywhere
ACCEPT all -- anywhere anywhere
REJECT all -- anywhere anywhere reject-with icmp-port-unreachable
REJECT all -- anywhere anywhere reject-with icmp-port-unreachable
REJECT all -- anywhere anywhere reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- anywhere anywhere state ESTABLISHED tcp spt:12443
ACCEPT tcp -- anywhere anywhere state ESTABLISHED tcp spt:openstack-id
ACCEPT tcp -- anywhere anywhere state ESTABLISHED tcp spt:amqps
ACCEPT tcp -- anywhere anywhere state ESTABLISHED tcp spt:5470
ACCEPT tcp -- anywhere anywhere state ESTABLISHED tcp spt:https
ACCEPT tcp -- anywhere anywhere state ESTABLISHED tcp spt:http
ACCEPT tcp -- anywhere anywhere state ESTABLISHED tcp spt:ssh
ACCEPT tcp -- anywhere anywhere state ESTABLISHED tcp spt:8428
ACCEPT tcp -- anywhere anywhere state ESTABLISHED tcp spt:8774
ACCEPT tcp -- anywhere anywhere state ESTABLISHED tcp spt:armtechdaemon
ACCEPT tcp -- anywhere anywhere state ESTABLISHED tcp spt:cslistener
ACCEPT tcp -- anywhere anywhere state ESTABLISHED tcp spt:9696
ACCEPT tcp -- anywhere anywhere state ESTABLISHED tcp spt:complex-main
ACCEPT tcp -- anywhere anywhere state ESTABLISHED tcp spt:8777
ACCEPT udp -- anywhere anywhere udp dpt:bootpc
```

Depending on an organization's security policy, these iptables firewall rules can be modified to enforce more strict access control policy. For example, a source IP address list can be defined to allow only specified IP address access to PowerVC services. This can be achieved by editing the `/etc/sysconfig/iptables` file and then refreshing the iptables service.

Example 6-5 shows the content of the default iptables rule file that is added by PowerVC.

Example 6-5 Rule file (/etc/sysconfig/iptables) added by PowerVC

```
# Generated by iptables-save v1.4.21 on Tue Nov  3 20:34:19 2015
*mangle
:PREROUTING ACCEPT [971866:1388944483]
:INPUT ACCEPT [971785:1388940137]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [364236:35872658]
:POSTROUTING ACCEPT [364260:35878429]
-A POSTROUTING -o virbr0 -p udp -m udp --dport 68 -j CHECKSUM --checksum-fill
COMMIT
# Completed on Tue Nov  3 20:34:19 2015
# Generated by iptables-save v1.4.21 on Tue Nov  3 20:34:19 2015
*nat
:PREROUTING ACCEPT [605:54671]
:INPUT ACCEPT [1:52]
:OUTPUT ACCEPT [237:18035]
:POSTROUTING ACCEPT [237:18035]
-A POSTROUTING -s 192.168.122.0/24 -d 224.0.0.0/24 -j RETURN
-A POSTROUTING -s 192.168.122.0/24 -d 255.255.255.255/32 -j RETURN
-A POSTROUTING -s 192.168.122.0/24 ! -d 192.168.122.0/24 -p tcp -j MASQUERADE -- to-ports
1024-65535
-A POSTROUTING -s 192.168.122.0/24 ! -d 192.168.122.0/24 -p udp -j MASQUERADE -- to-ports
1024-65535
-A POSTROUTING -s 192.168.122.0/24 ! -d 192.168.122.0/24 -j MASQUERADE
COMMIT
# Completed on Tue Nov  3 20:34:19 2015
# Generated by iptables-save v1.4.21 on Tue Nov  3 20:34:19 2015
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 12443 -j ACCEPT
-A INPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 35357 -j ACCEPT
-A INPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 5671 -j ACCEPT
-A INPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 5470 -j ACCEPT
-A INPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 443 -j ACCEPT
-A INPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 8428 -j ACCEPT
-A INPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 8774 -j ACCEPT
-A INPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 9292 -j ACCEPT
-A INPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 9000 -j ACCEPT
-A INPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 9696 -j ACCEPT
-A INPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 5000 -j ACCEPT
-A INPUT -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 8777 -j ACCEPT
-A INPUT -i virbr0 -p udp -m udp --dport 53 -j ACCEPT
-A INPUT -i virbr0 -p tcp -m tcp --dport 53 -j ACCEPT
-A INPUT -i virbr0 -p udp -m udp --dport 67 -j ACCEPT
-A INPUT -i virbr0 -p tcp -m tcp --dport 67 -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
```

```

-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 5901 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -d 192.168.122.0/24 -o virbr0 -m conntrack --ctstate RELATED,ESTABLISHED -j
ACCEPT
-A FORWARD -s 192.168.122.0/24 -i virbr0 -j ACCEPT
-A FORWARD -i virbr0 -o virbr0 -j ACCEPT
-A FORWARD -o virbr0 -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -i virbr0 -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
-A OUTPUT -p tcp -m state --state ESTABLISHED -m tcp --sport 12443 -j ACCEPT
-A OUTPUT -p tcp -m state --state ESTABLISHED -m tcp --sport 35357 -j ACCEPT
-A OUTPUT -p tcp -m state --state ESTABLISHED -m tcp --sport 5671 -j ACCEPT
-A OUTPUT -p tcp -m state --state ESTABLISHED -m tcp --sport 5470 -j ACCEPT
-A OUTPUT -p tcp -m state --state ESTABLISHED -m tcp --sport 443 -j ACCEPT
-A OUTPUT -p tcp -m state --state ESTABLISHED -m tcp --sport 80 -j ACCEPT
-A OUTPUT -p tcp -m state --state ESTABLISHED -m tcp --sport 22 -j ACCEPT
-A OUTPUT -p tcp -m state --state ESTABLISHED -m tcp --sport 8428 -j ACCEPT
-A OUTPUT -p tcp -m state --state ESTABLISHED -m tcp --sport 8774 -j ACCEPT
-A OUTPUT -p tcp -m state --state ESTABLISHED -m tcp --sport 9292 -j ACCEPT
-A OUTPUT -p tcp -m state --state ESTABLISHED -m tcp --sport 9000 -j ACCEPT
-A OUTPUT -p tcp -m state --state ESTABLISHED -m tcp --sport 9696 -j ACCEPT
-A OUTPUT -p tcp -m state --state ESTABLISHED -m tcp --sport 5000 -j ACCEPT
-A OUTPUT -p tcp -m state --state ESTABLISHED -m tcp --sport 8777 -j ACCEPT
-A OUTPUT -o virbr0 -p udp -m udp --dport 68 -j ACCEPT
COMMIT
# Completed on Tue Nov  3 20:34:19 2015

```

After iptables rules are modified, the iptables service must be restarted (as shown in Example 6-6) in order to apply the changed iptables rules.

Example 6-6 Restarting the iptables service

```

[root@p750_0_powervc /]# systemctl restart iptables
[root@p750_0_powervc /]#

```

Secure communication channels for PowerVC API endpoints

PowerVC exports public API services through the HTTPS protocol, by default. Accessing those services through the HTTP protocol is rejected. This improves the API security compared to the default OpenStack deployment, which allows the HTTP protocol for API access. However, internal access to the PowerVC API is through the HTTP protocol. For more information, see 6.3.2, “Secure communication for PowerVC APIs” on page 161.

Audit log and audit management tool

OpenStack implements API and security auditing that supports the DMTF Cloud Audit Data Federation standard to meet security and compliance requirements and also to more easily integrate with other security systems, such as SIEM.

PowerVC provides built-in support for collecting and retrieving audit data. It supports audit for compute (Nova), block storage (Cinder), networking (Neutron), image (Glance), metering (Ceilometer), and validation services. Due to a limitation in OpenStack’s audit middle ware, PowerVC currently lacks support for auditing the identity service (Keystone).

You can use the following PowerVC CLI commands:

powervc-config	Enables and disables auditing for services.
powervc-audit-export	Retrieves audit logs.

For more information about PowerVC’s auditing capabilities, see 6.4, “Audit” on page 162.

Note: If you experience performance issues, consider reducing the amount of additional audit data being collected.

Certification verification enforcement

PowerVC interacts with the Hardware Management Console (HMC), PowerVM NovaLink, SAN fabric switch, supported storage controllers, and PowerKVM for virtualization management. All these communication channels are encrypted with either SSL/TLS or SSH. However, a communication channel can potentially be compromised using a “man-in-the-middle” attack if PowerVC fails to verify the received certificate correctly. To avoid this security risk, PowerVC asks for verification of the certificate received from the external system by displaying the details instead of accepting the certificate without question.

For example, Figure 6-5 shows that a PowerVC user is asked to review and accept the certificate for the HMC when adding an HMC into the PowerVC realm. After the certificate is accepted as valid, it is stored in PowerVC. For future connections, the stored certificate can be used to verify that the certificate of the HMC has not been changed. If the stored certificate is not identical to the newly received certificate, PowerVC drops the connection. This indicates that the certificate might be forged to launch a man-in-the-middle attack.

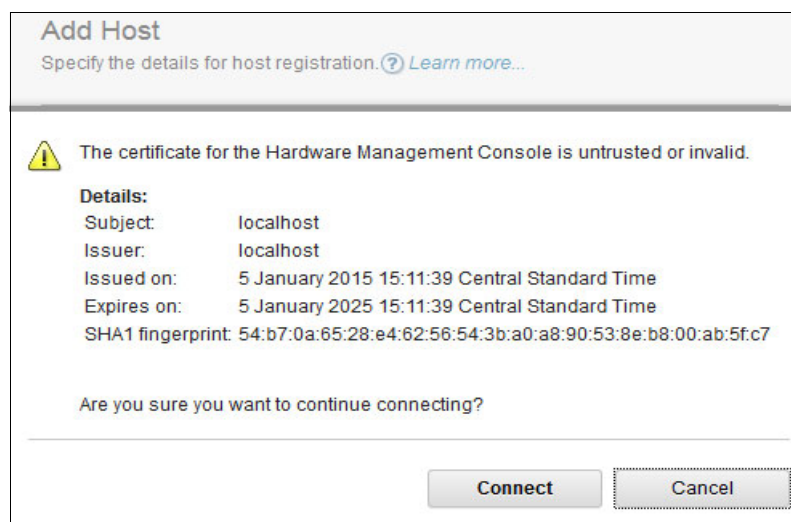


Figure 6-5 PowerVC asking for certificate validation

6.1.3 Secure communications

PowerVC is composed of several components that communicate with each other and also interact with external services provided by LDAP, HMC, PowerKVM hosts, and SAN switches. PowerVC provides REST API services so that cloud management solutions can support IBM Power Systems as cloud resources and communicate securely between all these components.

Figure 6-6 depicts how the components communicate internally and externally.

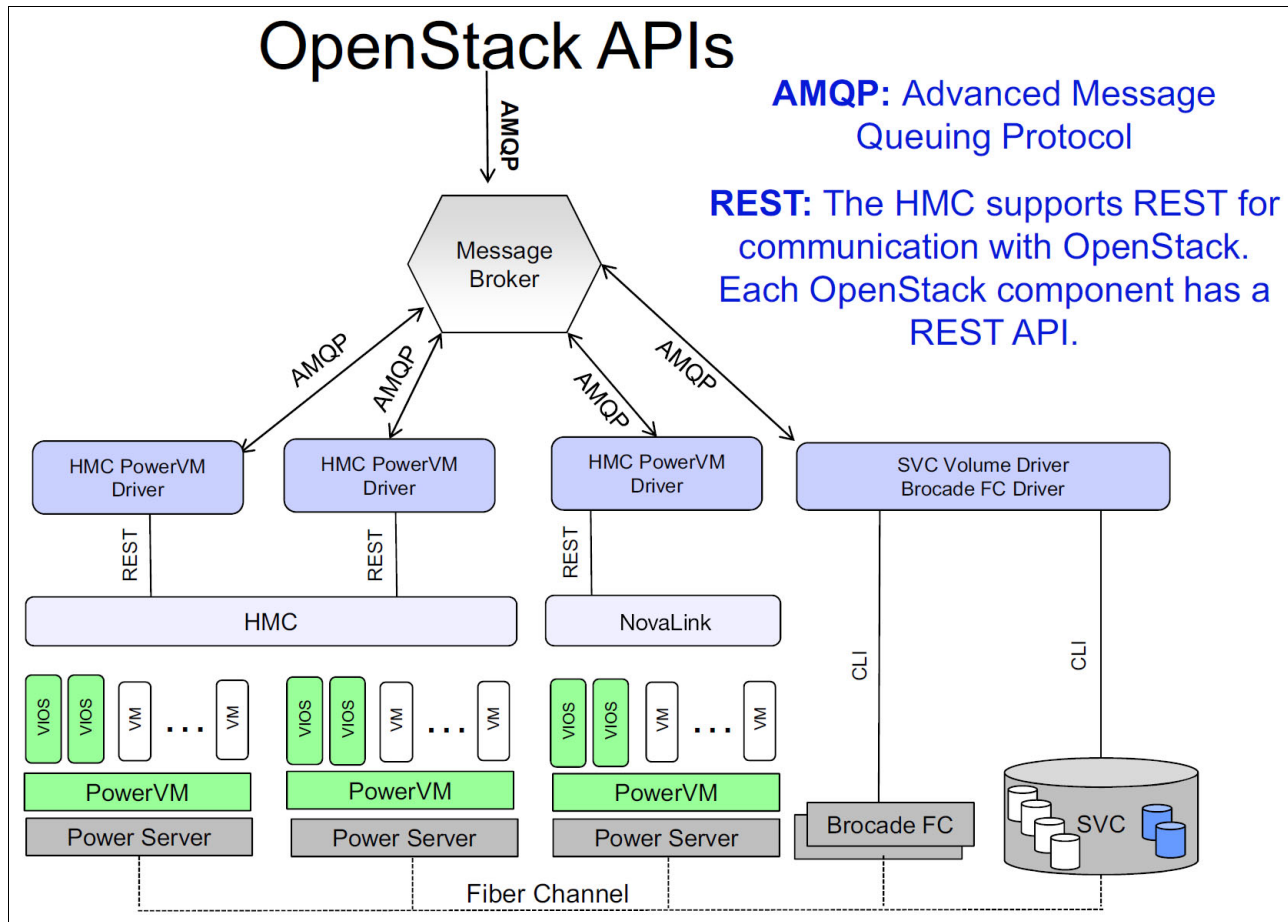


Figure 6-6 PowerVC architecture showing communication with external systems

Communication channels using SSL/TLS

To protect communication channels from attacks, such as eavesdropping or forgery, PowerVC uses either HTTPS or SSH, instead of using unencrypted and insecure protocols such as HTTP or Telnet.

The PowerVC message broker also uses AMQPS, which is an AMQP protocol over SSL/TLS. Also, PowerVC exposes its REST API services that use HTTPS only. PowerVC provides options to use LDAPS when it is configured to use LPAP directory for user authentication.

Table 6-5 maps the various services to the secure protocol.

Table 6-5 Secure protocols for each communication channel

Service	Secure protocol
PowerVC Web UI	HTTPS
PowerVC API (public and admin)	HTTPS
rabbitmq	AMQPS
Connection to HMC, PowerKVM, SAN switch, storage	SSH
Connection to LDAP	LDAPS or LDAP

To enable HTTPS, PowerVC creates key-pairs and self-signed certificates. After the installation is complete, a good suggestion is to replace these self-signed certificates with CA signed certificates.

Updating a self-signed certificate for Apache HTTP server

The PowerVC web UI and REST API access are secured by using HTTPS, provided by the Apache HTTP server. The self-signed X.509 certificate is created during PowerVC installation. This default certificate can be replaced with another self-signed certificate or a CA-signed certificate.

The Apache HTTP server uses the private key and certificate at these locations:

- ▶ Private key file: /etc/pki/tls/private/powervc.key
- ▶ Certificate file: /etc/pki/tls/certs/powervc.crt

To replace the certificate, follow these steps:

1. Replace the /etc/pki/tls/private/powervc.key file with a new private key file that has the same name.
2. Replace the /etc/pki/tls/certs/powervc.crt file with a new certificate key file that has the same name.
3. Restart PowerVC.

Updating a certificate for secure Live Partition Migration

Updating a certificate for secure Live Partition Migration is applicable only to PowerVC managing PowerKVM. Virtual machines that are managed by PowerKVM can be migrated to other PowerKVM managed hosts. For this, a communication channel is established by using TLS. The certificates used by these two communication channels must be replaced when they are expired or compromised.

Be aware that the certificate used for Live Partition Migration cannot be updated when any migration task is running.

Guidelines for the steps in the following procedure:

- ▶ If you want to keep the existing CA certificate but want to change a certificate signed by the CA certificate, skip to step 3. Instead of using a new CA certificate as described in the step, use the existing CA certificate.
- ▶ If you generate a new CA certificate, you must complete all of these steps.

Complete the following steps to update a certificate for secure Live Partition Migration:

1. Generate a new key file, which is optional, and generate a new self-signed CA certificate for the service. Replace the existing certificate and key, if generated, on the PowerVC management host. Use these file locations and names:
 - CA certificate: `/etc/pki/libvirt/certs/ca.crt`
 - CA private key: `/etc/pki/libvirt/private/ca.key`
2. Copy the new CA certificate onto each managed PowerKVM host. Use the following file location and name:
 - CA certificate: `/etc/pki/CA/cacert.pem`
3. Generate new certificates for PowerKVM hosts, as necessary. If you generated a new CA certificate in step 1, the certificates must be updated on all PowerKVM hosts. If you did not generate a new CA certificate, choose which hosts to update. Use the new CA certificate for the affected service to sign a new certificate for the managed KVM host. You can reuse the same private key or generate a new key.

Use these file locations and names:

- Server certificate: `/etc/pki/libvirt/servercert.pem`
 - Server private key: `/etc/pki/libvirt/private/serverkey.pem`
 - Client certificate: `/etc/pki/libvirt/clientcert.pem`
 - Client private key: `/etc/pki/libvirt/private/clientkey.pem`
4. Restart the relevant services on any host that has a new certificate:

```
# systemctl restart libvirtd.service
# systemctl restart openstack-nova-compute.service
```

6.2 Identity management

Identities in PowerVC are managed by the OpenStack Keystone service with modifications to use the security features of the operating system on the management host. The Keystone service handles authentication and authorization of user accounts.

You can configure PowerVC authentication to use an existing LDAP server as an identity repository anytime after PowerVC is installed.

The two types of users in PowerVC are Service users and PowerVC users:

- ▶ Service users are created during the PowerVC installation and they are used internally.
- ▶ PowerVC users are for the virtualization administrator to access the PowerVC web UI and PowerVC API services, and the root account is added to the PowerVC privileged group (admin) during PowerVC installation.

Note: PowerVC creates an additional operating system user account, `pwrvcdb`, which is used to manage the PowerVC internally used DB2 data repository and must never be modified or removed.

The following identity management topics are covered in this section:

- ▶ 6.2.1, “Removing the root account from the PowerVC admin group” on page 154
- ▶ 6.2.2, “PowerVC users, groups, roles, and policies” on page 155
- ▶ 6.2.3, “Using LDAP for PowerVC identity management” on page 156

6.2.1 Removing the root account from the PowerVC admin group

As a first cleanup task after PowerVC installation, create a PowerVC admin user account and remove the root account from the PowerVC admin group. PowerVC adds the root user into the PowerVC admin group without creating a separate admin account during installation. Initially, the PowerVC web UI is accessible only by using root account. If this change is not made, the root user account must always be used to access the PowerVC web UI, which is not a security best practice. You can use LDAP for user authentication.

Example 6-7 shows how to create a PowerVC admin account and assign the account to the admin group.

Example 6-7 Creating a PowerVC admin account

```
[root@p750_0_powervc /]# useradd -G admin pwrvcadmin
[root@p750_0_powervc /]# passwd pwrvcadmin
Changing password for user pwrvcadmin
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@p750_0_powervc /]# chfn -f "PowerVC Admin" pwrvcadmin
Changing finger information for pwrvcadmin
Finger information changed
[root@p750_0_powervc /]#
```

Example 6-8 shows how to remove the root account from the PowerVC admin group.

Example 6-8 Removing the root account

```
[root@p750_0_powervc /]# gpasswd -d root admin
Removing user root from group admin
```

The change is immediately applied to PowerVC and the change can also be validated using PowerVC web UI. Figure 6-7 shows the user list after removing root and adding a new PowerVC admin user. Be aware that email information cannot be assigned when PowerVC uses the operating system user management feature instead of LDAP.

Viewing only: The PowerVC web UI allows only *viewing* of users and groups. Users and groups cannot be created or deleted by using the PowerVC web UI.

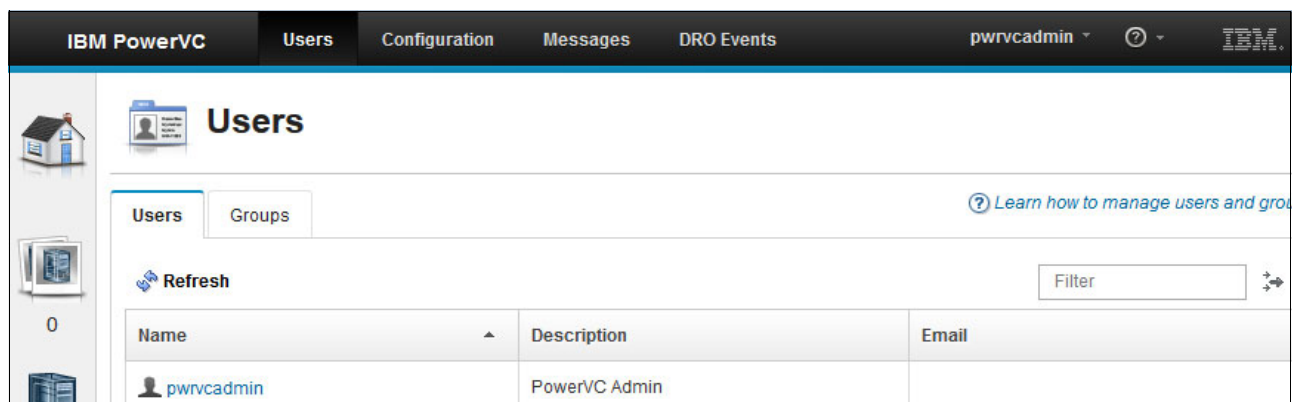


Figure 6-7 PowerVC user list

6.2.2 PowerVC users, groups, roles, and policies

The PowerVC user ID is used to verify access to the PowerVC web UI or API services. The default installation of PowerVC uses the account management feature of the operating system. Any modification to `/etc/passwd` and `/etc/group` is instantly reflected within the PowerVC security settings.

Users and groups in PowerVC are maintained within the operating system's user and group realm unless LDAP is configured to manage them. Roles and policies are always maintained by PowerVC configuration files. Groups, roles, and policies for each PowerVC service are predefined. The predefined roles should not be modified.

PowerVC defines three user groups in `/etc/group`:

admin	Members are granted admin role.
deployer	Members are granted deployer role.
viewer	Members are granted viewer role.

Three roles of PowerVC are as follows:

admin	Members can perform all tasks to all resources.
deployer	Same as admin role, except members cannot add, update, or remove storage systems, hosts, networks, and storage templates. Members cannot add or remove existing virtual machines, nor can they add or remove existing volumes to be managed by PowerVC. Members cannot update fibre channel ports, nor can they view users and groups.
viewer	Members can view resource information, but cannot perform tasks. Members are restricted from viewing users and groups.

The **powervc-keystone** PowerVC CLI command can list users, groups, and roles, and assign a role to users and groups. This command can also remove roles from users and groups. However, to create or remove users you must use the operating system user management commands **useradd**, **usermod**, or **userdel**.

Example 6-9 shows how to list roles by using the **powervc-keystone** PowerVC CLI command.

Example 6-9 Using the powervc-keystone command to list roles

```
[root@p750_0_powervc ~]# powervc-keystone list-roles
Password
id          9215afd81e1d4b84b653dac43ce65041
name        admin

id          00c4b8c315294b13b57d0d6a54cc60a9
name        deployer

id          ece28a5218db4131a5602d01dbe90ebc
name        viewer
```

Each service defines its own policy to assign permissions to roles. The policy is defined in `policy.json` files for each service.

PowerVC defines only two projects or tenants and only one domain, referred to as *default*. The two projects are as follows:

ibm-default	Project for all users
service	Project for OpenStack service users

Disabling remote access to the management system

When the operating system user repository is used to handle PowerVC identities, disable the remote login capability for all users. This step is important because PowerVC users need to access only the PowerVC web UI or PowerVC API; there is no reason to log in to the PowerVC server operating system except for the PowerVC administrator. You can disable remote login by changing the user's shell program to `/sbin/nologin`.

Example 6-10 shows how to create a user in the deployer group and disable the remote login capability.

Example 6-10 Creating a PowerVC user with remote login disabled

```
[root@p750_0_powervc ~]# useradd -g deployer pwrvcdeployer
[root@p750_0_powervc ~]# passwd pwrvcdeployer
Changing password for user deployer.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@p750_0_powervc ~]# usermod -s /sbin/nologin pwrvcdeployer
```

Configuration step not needed: If LDAP or Microsoft Active Directory is used for identity management, this configuration step is not needed.

6.2.3 Using LDAP for PowerVC identity management

By default, PowerVC users and groups are managed by the operating system. To enhance identity management, PowerVC can be configured to work with your LDAP server anytime after installation by using the `powervc-ldap-config` command. The command can also be used to revert to using the operating system to manage users and groups instead of using your LDAP server. User accounts are not deleted even after the change is made.

During LDAP configuration, the default is to use TLS because using an unencrypted channel for LDAP access is not secure.

For more information about the PowerVC and `powervc-ldap-config` options, see the IBM Knowledge Center:

- ▶ For managing PowerVM:
http://www.ibm.com/support/knowledgecenter/SSXK2N_1.3.0/com.ibm.powervc.standard.help.doc/powervc_ldap_hmc.html
- ▶ For managing PowerKVM:
http://www.ibm.com/support/knowledgecenter/SSXK2N_1.3.0/com.ibm.powervc.kvm.help.doc/powervc_ldap_kvm.html

Example 6-11 shows how to configure PowerVC to use LDAP without TLS. Notice the warning in this example.

Example 6-11 Changing PowerVC to use LDAP

```
[root@p750_0_powervc openldap]# powervc-ldap-config
Configuring PowerVC for LDAP.
URL [ldap://localhost]:172.31.03.04
Use TLS [y]: n
Warning: LDAP communication will not be secured!
Continue without TLS [n]:
```

```

Anonymous bind [n]:
User tree DN [ou=Users,dc=example,dc=com]: ou=Users,dc=ibm,dc=com
User filter [None]:
User object class [inetOrgPerson]:
User ID attribute [uid]:
User name attribute [cn]:
User mail attribute [email]: mail
User password attribute [userPassword]:
User description attribute [description]:
Group tree DN [ou=Groups,dc=example,dc=com]: ou=Groups,dc=ibm,dc=com
Group filter [None]:
Group object class [groupOfNames]:
Group ID attribute [cn]:
Group name attribute [cn]:
Group member attribute [member]:
Group description attribute [description]:
Restarting keystone: [ OK ]
Group for admin role users [admin]:
Group for deployer role users [deployer]:
Group for viewer role users [viewer]:
Updating /etc/keystone/domains/keystone.Default.conf
Removing previous role grants
Restarting keystone: [ OK ]
Adding role grants
Adding 'admin' role for group 'admin':
Adding 'deployer' role for group 'deployer':
Adding 'viewer' role for group 'viewer':
Restarting keystone: [ OK ]

```

After the switch to LDAP is completed, the `/etc/keystone/domains/keystone.Default.conf` file is updated to reflect LDAP connectivity information in the `[ldap]` section, and the `[identity]` section shows that the driver is set to `ldap` instead of `local`.

PowerVC can be converted back to using the operating system user repository with the **powervc-ldap-config** command and the **-s off** flag as shown in Example 6-12.

Example 6-12 Switching back to the operating system user repository

```

[root@p750_0_powervc openldap]# powervc-ldap-config -s off
Configuring PowerVC for local OS.
Restarting keystone: [ OK ]
Group for admin role users [admin]:
Group for deployer role users [deployer]:
Group for viewer role users [viewer]:
Removing previous role grants
Restarting keystone: [ OK ]
Adding role grants
Adding 'admin' role for group 'admin':
Adding 'deployer' role for group 'deployer':
Adding 'viewer' role for group 'viewer':
Restarting keystone: [ OK ]

```

6.3 API security

The OpenStack software has industry-standard interfaces that are released under the terms of the Apache License. PowerVC interfaces are a subset of the OpenStack northbound APIs. A number of interfaces were added or extended to enhance the capabilities that are associated with the IBM Power platform. Several types of APIs can be used to automate virtualized resource management. Note that JSON should be used as interchange format when using these APIs.

The three types of PowerVC APIs are as follows:

- ▶ Supported OpenStack APIs

PowerVC provides a subset of OpenStack APIs that can be used without any modification.

- ▶ Extended OpenStack APIs

These APIs are also a subset of OpenStack APIs, but are extended to support PowerVC functions.

- ▶ PowerVC APIs

These APIs are only for PowerVC.

For API lists and usages for PowerVC 1.3., see the IBM Power Virtualization Center Standard Version 1.3 documentation in the IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSXK2N_1.3.0

The following API security topics are covered in this section:

- ▶ 6.3.1, “Authentication” on page 158
- ▶ 6.3.2, “Secure communication for PowerVC APIs” on page 161
- ▶ 6.3.3, “Strict network access control” on page 162

6.3.1 Authentication

To invoke PowerVC APIs, users first must be authenticated by PowerVC Keystone using a proper credential. After a user is authenticated, PowerVC returns a unique token for the current session that by default expires after 6 hours. Using this token, users can send API requests to PowerVC endpoints by using the HTTPS protocol. Figure 6-8 depicts how the authentication and API request and response are performed.

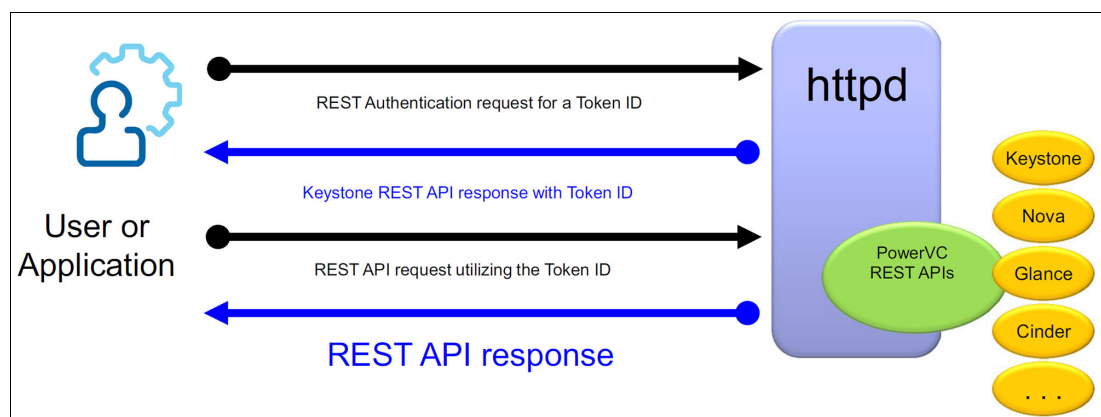


Figure 6-8 How the authentication and API request and response are performed

This section covers the following authentication topics:

- ▶ “Setting the expiration interval for authentication tokens” on page 159
- ▶ “Warning a user before authentication token expires” on page 160
- ▶ “Setting UI inactivity time out” on page 161

Setting the expiration interval for authentication tokens

In OpenStack, the three most common types of authentication tokens are *UUID token*, *PKI token*, and *fernet token*.

PowerVC supports the UUID token which expire by default after 6 hours. This can be changed by using the **powervc-config identity token_expiration** PowerVC CLI command.

Example 6-13 shows the output of the **powervc-config identity token_expiration** command without any additional parameters.

Example 6-13 Viewing the token expiration value

```
[root@p750_0_powervc ~]# powervc-config identity token_expiration
Current value: 6:00:00
Default value: 6:00:00
```

Example 6-14 shows how to set the expiration time to 8 hours. If you do not specify the **--unit** parameter for the time unit (which is in seconds by default) you do this by specifying 28800 including the **--restart** parameter in order for the change to take effect.

Example 6-14 Setting the token to expire after 8 hours using seconds (the default units)

```
[root@p750_0_powervc ~]# powervc-config identity token_expiration --set 28800
--restart
Setting token_expiration to 28800 sec
Restarting the httpd service
[root@p750_0_powervc ~]# powervc-config identity token_expiration
Current value: 8:00:00
Default value: 6:00:00
```

The command shown in Example 6-15 does the same as the command in Example 6-14, but uses the **--unit** parameter set to the time unit to hours.

Example 6-15 How to set token to expire after 8 hours specifying hours and viewing

```
[root@p750_0_powervc ~]# powervc-config identity token_expiration --set 8 --unit
hr --restart
Setting token_expiration to 8 hr
Restarting the httpd service
[root@p750_0_powervc ~]# powervc-config identity token_expiration
Current value: 8:00:00
Default value: 6:00:00
```

Warning a user before authentication token expires

In addition to the expiration interval, you can control the time to warn a user before the user's authentication token expires. The web console prompts the user to re-enter the password, by which the user obtains a new authentication token. This can help users avoid starting operations that might fail due to an expired token before the operation is completed. The default is 15 minutes as shown in Example 6-16.

Example 6-16 Displaying a web token expiration warning time

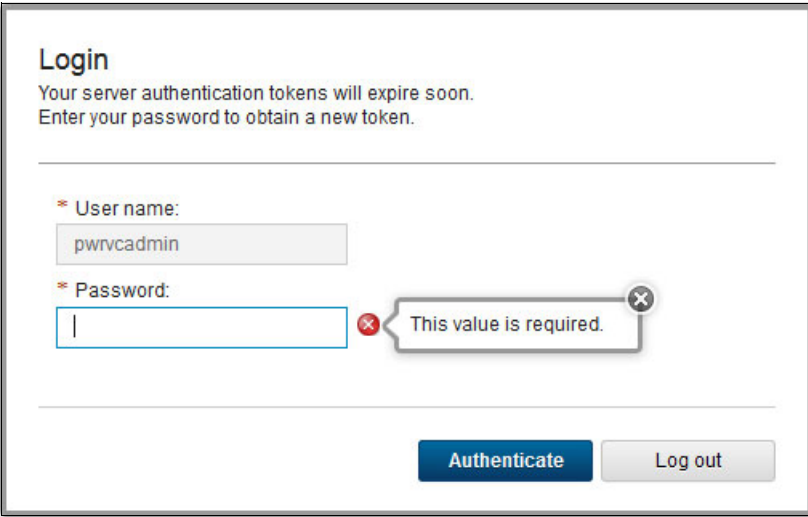
```
[root@p750_0_powervc ~]# powervc-config web token_expiration_warning_time  
Current value: 0:15:00  
Default value: 0:15:00
```

To change the current value, you use the syntax shown in Example 6-17. Note that the change does not need a restart, takes effect immediately, and works for new tokens. Therefore if you are already signed on, you will need to sign off and sign on again.

Example 6-17 Setting the web token expiration warning time using minutes which is the default unit

```
[root@p750_0_powervc ~]# powervc-config web token_expiration_warning_time --set 60  
Setting token_expiration_warning_time to 60 min  
[root@p750_0_powervc ~]# powervc-config web token_expiration_warning_time  
Current value: 1:00:00  
Default value: 0:15:00
```

When the warning time is reached, the PowerVC user receives the notification message shown in Figure 6-9.



The screenshot shows a web browser window titled "Login". Below the title, a message states: "Your server authentication tokens will expire soon. Enter your password to obtain a new token." There are two input fields: "User name:" with the text "pwrvcadmin" and "Password:" which is currently empty. A red error message bubble with a close button (X) points to the password field, containing the text "This value is required." At the bottom right, there are two buttons: "Authenticate" (in blue) and "Log out" (in grey).

Figure 6-9 Expiration notification window

Setting UI inactivity time out

You can also control the time a user is allowed to idle on the PowerVC web console before being prompted to confirm that they are still active. The default value is 2 hours. To display the current value, use the syntax shown in Example 6-18.

Example 6-18 Displaying web inactivity time out

```
[root@p750_0_powervc ~]# powervc-config web inactivity_timeout  
Current value: 2:00:00  
Default value: 2:00:00
```

To change the web inactivity time out value, use the syntax shown in Example 6-19. Note that this change also does not need a restart and takes effect immediately.

Example 6-19 Changing web inactivity time out to 15 minutes which is the default units

```
[root@p750_0_powervc ~]# powervc-config web inactivity_timeout --set 15  
Setting inactivity_timeout to 15 min  
[root@p750_0_powervc ~]# powervc-config web inactivity_timeout  
Current value: 0:15:00  
Default value: 2:00:00
```

When the inactivity time is reached, the PowerVC user receives the notification message shown in Figure 6-10.

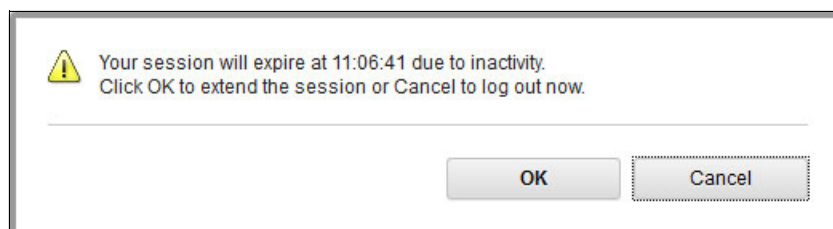


Figure 6-10 Inactivity time reached on PowerVC UI

6.3.2 Secure communication for PowerVC APIs

As the OpenStack Security Guide⁵ mentions, having a secure communication channel for API services is important. Only OpenStack Keystone provides a method to configure this channel as SSL enabled. For other API services, OpenStack leaves secure communication configuration to the deployer. PowerVC implements an HTTPS reverse proxy to provide secure API services, so that the API requests are sent to the HTTPS proxy instead of directly to API service eventlets. Figure 6-11 shows how an API client connects to PowerVC API through an HTTPS proxy.

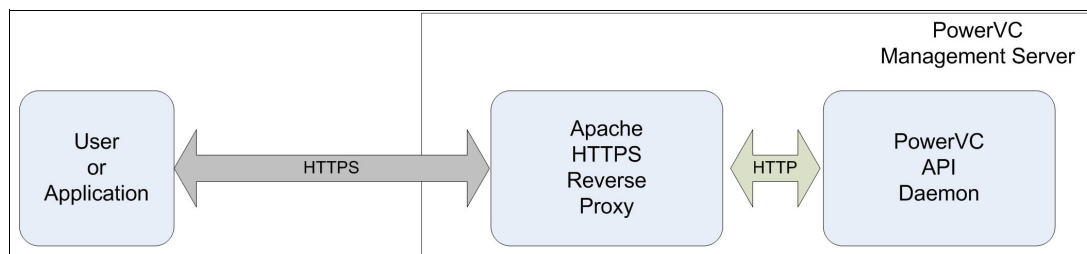


Figure 6-11 API client connecting to PowerVC API through HTTPS proxy

⁵ <http://docs.openstack.org/security-guide/content/>

Each API service daemon has two listening ports. One port uses HTTP for local connections and the other port uses HTTPS for remote connections. Any external users and applications must send API requests using the HTTPS protocol.

For example, the Nova API service daemon has two listening ports as Example 6-20 shows. The first port accepts HTTP only from local hosts and the second port accepts HTTPS connections from remote hosts. This is acceptable because the local connection within the host is considered secure.

Example 6-20 Nova API listening ports

tcp	0	0	127.0.0.1:8774	0.0.0.0:*	LISTEN
tcp	0	0	172.16.22.13:8774	0.0.0.0:*	LISTEN

Using these ports, admin and public endpoints are accessible only through HTTPS protocols and internal endpoints allow HTTP protocols.

6.3.3 Strict network access control

By default, PowerVC installs iptables firewalls rules unless it is instructed not to. The rules are defined to accept PowerVC API services requests and web UI access from any IP address. All other unnecessary ports that are not needed for running PowerVC are blocked.

In this approach, any user or application with valid credentials can potentially access PowerVC APIs. However, this approach does not adhere to a well-known security principle, *least privilege*. Therefore, a suggested approach is to restrict source IP addresses that can access PowerVC API endpoints or the web UI.

This restriction can be achieved in several ways. If a network firewall is in front of PowerVC, that firewall can be used to restrict network access to PowerVC. The simplest way is to define rules to allow connections to PowerVC through port 443/TCP from a trusted IP list because all API access from remote users go through the HTTPS reverse proxy server listening on port 443/TCP.

Restrictions: The PowerVC web UI and API services are all accessed through port 443/TCP (HTTPS). Therefore, preventing API users from accessing the PowerVC web UI is not possible. However, this does not introduce any security risk because the same policy on functions is applied to the user account.

6.4 Audit

To ensure a good security posture in any area, *visibility* must be achieved. Without knowing what is happening in your environment, it is hard to protect anything against attacks and respond quickly when an attack is occurring. Besides being an essential ingredient for security, auditing is the cornerstone to meet various security compliance requirements.

To meet these needs, an audit feature was added to OpenStack Ceilometer. This audit feature supports the DMTF Cloud Auditing Data Federation standard, which defines how to use common audit data among various cloud services.

6.4.1 Enabling and disabling PowerVC audit

Use the PowerVC CLI command, **powervc-config general audit**, to view the audit status for all PowerVC services.

To view the audit status for a specific PowerVC service (compute, image, metering, network, storage, or validation) use the **powervc-config <service> audit** command.

To enable auditing for a PowerVC service use the **--enable** flag with the command and use the **--disable** flag to disable auditing. Use the **--restart** flag for changes to take effect.

Restriction: To start, stop, or check the audit status, **powervc-config** must be run as root.

Example 6-21 shows how to check the audit status and how to enable and disable audit for a specific PowerVC service by using the **powervc-config** command.

Example 6-21 powervc-config command usage to view and control the audit feature

```
[root@p750_0_powervc ~]# powervc-config general audit
compute auditing is DISABLED with ignore list "GET"
image auditing is DISABLED with ignore list "GET"
metering auditing is DISABLED with ignore list "GET"
network auditing is DISABLED with ignore list "GET"
storage auditing is DISABLED with ignore list "GET"
validation auditing is DISABLED with ignore list "GET"

[root@p750_0_powervc ~]# powervc-config network audit --enable --restart
Restarting network services

[root@p750_0_powervc ~]# powervc-config network audit
network auditing is ENABLED with ignore list "GET"

[root@p750_0_powervc ~]# powervc-config network audit --disable --restart
Restarting network services

[root@p750_0_powervc ~]# powervc-config network audit
network auditing is DISABLED with ignore list "GET"
```

6.4.2 Retrieving audit log information

OpenStack Ceilometer (telemetry service) stores audit data in a DB2 database and provides REST APIs for flexible querying. PowerVC users in the admin group can retrieve audit log information by using the **powervc-audit-export** command.

The options for the **powervc-audit-export** command are listed in Table 6-6.

Table 6-6 Command-line options for powervc-audit-export command

Options	Comment
<ul style="list-style-type: none">▶ -u <user name>▶ --user_name <user name>	Either option tells the powervc-audit-export command to send a request as a specific user. If not provided, the logged-in user name is used.
<ul style="list-style-type: none">▶ -n <number of records>▶ --top_n <number of records>	Upper bound for number of audit records to return. The default for this optional flag is 1000.

Options	Comment
<ul style="list-style-type: none"> ▶ <code>-o <output file></code> ▶ <code>--output <output file></code> 	Either option specifies the file name to export audit data. The defaults for this optional flag are <code>export_audit.json</code> or <code>export_audit.csv</code> depending on the output format.
<ul style="list-style-type: none"> ▶ <code>-f <filter file></code> ▶ <code>--filter <filter file></code> 	The file containing filter parameters with which to further control what audit data is exported, for example, to get data for a single user. For more information about filter structure and usage, see the PowerVC topic in the IBM Knowledge Center. This flag is optional.
<ul style="list-style-type: none"> ▶ <code>-x {json,csv}</code> ▶ <code>--output_format {json,csv}</code> 	Using either option, the format of output can be selected. The supported formats are <code>json</code> and <code>csv</code> . The default is <code>json</code> for this optional flag.

Example 6-22 shows a command to export audit data from the audit repository into a file. By default, this data is exported in JSON format, which adheres to the DMTF CADF standard.

Example 6-22 Exporting audit data into a file in JSON format

```
[pwrvcadmin@p750 ~]$ powervc-audit-export -o ./powervc-audit-nov06.out
Password
Audit records written to file: ./powervc-audit-nov06.out
```

Filtering audit log

PowerVC audit logs can be filtered with the **powervc-audit-export** command by using a filter file. The filter can be defined using a JSON object name, an operator, and a value.

For example, to filter the audit log for nova-api service entries where the requestor name is root, create a filter file with the filter definition that is shown in Example 6-23.

Example 6-23 Audit filter to filter audit log with root as the initiator

```
[
  {
    "q.field": "service",
    "q.op": "eq",
    "q.value": "nova-api"
  },
  {
    "q.field": "initiator_name",
    "q.op": "eq",
    "q.value": "root"
  }
]
```

Besides the `initiator.name` object, any object, such as event time, event activity, and so on, can be used to filter the audit event log. You can generate an audit log file using the **powervc-audit-export** command without any option and check the generated audit log file for other object names.

6.4.3 Important log files

Other security-related log files are collected in the following directories:

- ▶ /var/log/keystone
- ▶ /var/log/httpd

For example, the /var/log/keystone/keystone.log indicates failed authentication attempts from the local host as shown in Example 6-24.

Example 6-24 Authentication failure log in keystone.log file

```
2015-11-11 10:12:36.356 14757 WARNING keystone.common.wsgi [-] Authorization
failed. The request you have made requires authentication. from ::1
```

6.5 Security options using powervc-config command

A number of security options can be managed by the **powervc-config** command. The auditing features are explained in 6.4.1, “Enabling and disabling PowerVC audit” on page 163; tokens are explained in 6.3.1, “Authentication” on page 158.

This section describes two options that are relevant to limiting the usage of storage capacity, which can help in preventing someone from maliciously filling up your disk space. You should determine an appropriate value for your environment and change the settings.

6.5.1 Setting the maximum image size

The first option is related to the maximum image size that can be uploaded through the image service.

To display the current value and default value, which is set to 1 TB, use the command shown in Example 6-25 where it is shown as 1099511627776 bytes.

Example 6-25 Displaying the current and default value for the maximum image size in Bytes

```
[root@p750_0_powervc ~]# powervc-config image image_size_cap
Current value: 1099511627776 B
Default value: 1099511627776 B
```

You can use the **--unit** parameter to see the image size in the unit of your preference as shown in Example 6-26.

Example 6-26 Displaying the current and default value for maximum image size in TB

```
[root@p750_0_powervc ~]# powervc-config image image_size_cap --unit TB
Current value: 1 TB
Default value: 1 TB
```

For example, to change the maximum value to 8 TB, use the command shown in Example 6-27. Notice that you must restart the image service.

Example 6-27 Setting the maximum image size to 8TB

```
[root@p750_0_powervc ~]# powervc-config image image_size_cap --set 8 --unit TB --restart
Setting image_size_cap to 8 TB
Restarting the image service
[root@p750_0_powervc ~]# powervc-config image image_size_cap --unit TB
Current value: 8 TB
Default value: 1 TB
```

6.5.2 Setting the maximum amount of per-user image storage

This option is the maximum amount of image storage that each user can use across all storage systems. This value is system-wide, it cannot be configured on a per user basis.

To display the current value and default value, which is set to 0 bytes (which means unlimited), you use the command shown in Example 6-28.

Example 6-28 Display the current and default value for maximum storage amount which is unlimited

```
[root@p750_0_powervc ~]# powervc-config image user_storage_quota
Current value: 0 B
Default value: 0 B
```

For example, to change the, to change the maximum value to 80 GB, use the command shown in Example 6-29. You can use the `--unit` parameter to see it in the unit of your preference as shown in the second command in Example 6-29. Notice that you must restart the image service.

Example 6-29 Setting the amount per user image storage limit to 80GB

```
[root@p750_0_powervc ~]# powervc-config image user_storage_quota --set 80 --unit GB --restart
Setting user_storage_quota to 80GB
Restarting the image service
[root@p750_0_powervc ~]# powervc-config image user_storage_quota --unit GB
Current value: 80 GB
Default value: 0 GB
```

6.6 Patch management

To prevent attacks that exploit known vulnerabilities, maintain a PowerVC environment that is up-to-date so that all known security vulnerabilities are properly and promptly addressed. Be sure to consider the following components because you are responsible for keeping them up-to-date:

- ▶ Red Hat Enterprise Linux server
- ▶ PowerVC
- ▶ Apache HTTP server
- ▶ OpenSSL
- ▶ OpenSSH

Note: Because the DB2 and OpenStack drivers are included with PowerVC, the patches to these components are packaged as PowerVC fix packs or iFixes.

6.6.1 Where to get PowerVC security patch information

An IBM security bulletin lists security vulnerabilities with appropriate fixes, if available. See the IBM Support Portal:

<http://www.ibm.com/support>

Sign in to the Support Portal and use the **Product finder** field to search for PowerVC. Select **Flashes, alerts and bulletins** (under Product Support Content) to check for current updates.

A suggestion is to subscribe to **support notifications**, which you can do by selecting **Manage your support notifications** (under Tools and resources). On that page, find PowerVC again, and subscribe. This way you will receive automatic email notifications about new security-related news and vulnerabilities.

6.6.2 Consideration on OpenStack vulnerability

Because PowerVC 1.3 is built on the OpenStack Liberty release, any security vulnerability found in OpenStack can potentially affect PowerVC also. The following OpenStack components are used by PowerVC:

- ▶ Keystone: Identity service
- ▶ Nova: Compute service
- ▶ Neutron: Network service
- ▶ Cinder: Block storage service
- ▶ Glance: Image management service
- ▶ Ceilometer: Monitoring and event service

Note: Security patches for these OpenStack components are provided in PowerVC fix packs or iFixes.

6.6.3 Managing Open Source components like Apache HTTP server, OpenSSL, and OpenSSH

Besides OpenStack Liberty components, PowerVC uses components like the Apache HTTP server, OpenSSL, and OpenSSH to name a few. Be sure to watch for any security vulnerability of these products.

You can check your current Apache HTTP server version, as shown in Example 6-30.

Example 6-30 Checking the version of the Apache HTTP server

```
[root@p750_0_powervc ~]# /usr/sbin/httpd -version
Server version: Apache/2.4.6 (Red Hat Enterprise Linux)
Server built:   Dec  2 2014 08:09:42
```

In addition, you must monitor any vulnerabilities for the **openssl** and **openssh** packages because PowerVC secures communication channels using SSL/TLS and SSH from the **openssl** and **openssh** packages.

To find the OpenSSL and OpenSSH version numbers, use the **ssh -V** command as shown in Example 6-31.

Example 6-31 Checking the version of OpenSSL and OpenSSH

```
[root@p750_0_powervc ~]# ssh -V
OpenSSH_6.6.1p1, OpenSSL 1.0.1e-fips 11 Feb 2013
```

For OpenSSL vulnerability information, see the following web page:

<http://www.openssl.org/news/vulnerabilities.html>

For OpenSSH vulnerability information, see the following web page:

<http://www.openssh.com/security.html>

6.7 Conclusion

Because PowerVC is designed with security from the beginning, and because the default installation already enforces many security features, not many extra steps must be done to secure the PowerVC environment. However, you must do several tasks to make PowerVC even more secure, or to reflect the organization's individual security policy. Use the following checklist for PowerVC security:

- ▶ Enable PowerVC audit.
- ▶ Enable iptables firewall.
- ▶ Create a PowerVC admin user and remove root from the admin group.
- ▶ Consider changing PowerVC to use LDAP for authentication when you have that available in your organization.
- ▶ Replace certificates for the Apache HTTP server with self-signed or CA signed certificates.
- ▶ Harden the PowerVC management server, secure the operating system, and do not install any other software.
- ▶ Keep your system up-to-date with security patches for PowerVC and other supporting components.



IBM Cloud Manager with OpenStack security

IBM Cloud Manager with OpenStack is an easy-to-deploy and simple-to-use cloud management software offering. It is based on OpenStack with IBM enhancements that feature a self-service portal for workload provisioning, virtual image management, and monitoring. It is an innovative, cost-effective approach that also includes automation, metering, and security for your virtualized environment.

The information in this chapter is based on IBM Cloud Manager with OpenStack V4.3.0, which is built on OpenStack Kilo.

The following topics are covered in this chapter:

- ▶ 7.1, “Introducing IBM Cloud Manager with OpenStack” on page 170
- ▶ 7.2, “Identity” on page 173
- ▶ 7.3, “Access” on page 177
- ▶ 7.4, “Patch management” on page 187
- ▶ 7.5, “Audit and logging” on page 188
- ▶ 7.6, “Image management” on page 188
- ▶ 7.7, “REST API security” on page 189
- ▶ 7.8, “Conclusion” on page 190

7.1 Introducing IBM Cloud Manager with OpenStack

By installing the IBM Cloud Manager with OpenStack product, you receive the tools to deploy an OpenStack cloud, consisting of an OpenStack controller node and one or more compute hosts.

This section covers the following topics:

- ▶ 7.1.1, “OpenStack and Chef” on page 170
- ▶ 7.1.2, “Enhancements to OpenStack” on page 170
- ▶ 7.1.3, “Power Systems hypervisor support” on page 171
- ▶ 7.1.4, “Deployment models” on page 172

7.1.1 OpenStack and Chef

OpenStack is a collection of open source technology projects cosponsored by a broad group of industry leaders, including IBM. It provides an operating platform for managing clouds on a massive scale. Its technology is hypervisor independent and includes software to provision virtual machines (VMs) on standard hardware. In addition, it offers a distributed object store and a wide range of optional functionality, including a network controller, authentication manager, management dashboard, and block storage.

At the time of writing, the most current OpenStack release includes the following components:

- ▶ Compute (Nova)
- ▶ Network (Neutron)
- ▶ Block Storage (Cinder)
- ▶ Object Storage (Swift)
- ▶ Authentication (Keystone)
- ▶ Image (Glance)
- ▶ Metering (Ceilometer)
- ▶ Orchestration (Heat)

IBM Cloud Manager with OpenStack provides full access to the OpenStack Kilo APIs and is also extensible through the REST API.

OpenStack also incorporates Chef technology to provide a robust installation and configuration method by using *cookbooks*, *recipes*, *environments*, and *roles*. Chef provides a client that communicates with the deployment server to also install software on remote nodes.

OpenStack documentation is available at the following website:

<http://docs.openstack.org>

Parts that are relevant to IBM Cloud Manager with OpenStack security are referred to in this document.

7.1.2 Enhancements to OpenStack

IBM provides several enhancements to OpenStack in the IBM Cloud Manager with OpenStack offering.

In addition to the OpenStack dashboard interface (Horizon module) that is intended for cloud administrators, IBM Cloud Manager also provides a self-service portal, intended for management and cloud users.

With the self-service portal, you can do the following public and private cloud operations:

- ▶ Provision and deprovision virtual servers on OpenStack (KVM, PowerKVM, Hyper-V, PowerVC, z/VM) and VMware vSphere using vCenter virtualization environments.
- ▶ Provide access to multiple clouds from a single portal.
- ▶ Allow creation of expiration policies to reduce the numbers of unused virtual machines.
- ▶ Support for deploy, resize and capture, backup and restore, image management, approvals, expiration, billing, and metering.
- ▶ Allow creation of projects to grant team-specific access to instances.
- ▶ Support for PowerKVM including volume management, flavor management, secure shell (SSH) key management, and basic multitenancy support.

IBM Cloud Manager also provides world-class support from IBM, and a hybrid capability to integrate with SoftLayer® through an IBM Services offering.

The IBM Cloud Manager self-service portal uses the following FIPS 140-2 approved cryptographic providers:

- ▶ IBMJCEFIPS (certificate 376)
- ▶ IBMJSSEFIPS (certificate 409)
- ▶ IBM Crypto for C (certificate 384)

The IBM Platform Resource Scheduler (PRS) has been integrated into IBM Cloud Manager 4.3 for advanced scheduling capabilities. Advantages for using PRS include higher quality of service, improved performance, greater flexibility, automated management, and reduced operating costs. It is the default scheduler for all topologies, except for the *HA controller +n compute* topology. An overview of topologies is listed in 7.1.4, “Deployment models” on page 172.

IBM Cloud Manager with OpenStack also allows the use of an enterprise-level database in DB2, and the ability to migrate to the full-featured IBM Cloud Orchestrator offering.

At the time of writing this book, the version of IBM Cloud Manager with OpenStack is V4.3. For IBM Cloud Manager with OpenStack V4.3 product documentation, see the IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SST55W_4.3.0

7.1.3 Power Systems hypervisor support

IBM Cloud Manager supports the IBM PowerKVM hypervisor and the IBM PowerVM hypervisor through IBM PowerVC. PowerKVM compute nodes must be hosts. Nested virtual machines are not supported.

PowerKVM management: PowerKVM hosts are managed directly by IBM Cloud Manager with OpenStack, and not through PowerVC.

The PowerKVM host must satisfy the following requirements:

- ▶ Operating system: IBM_PowerKVM release 2.1.0, 2.1.1, and 3.1
- ▶ Hardware: POWER8 server with the PowerKVM hypervisor configured

IBM Cloud Manager is compatible with IBM Power Virtualization Center (PowerVC) Standard version 1.2.0.1, 1.2.0.2, 1.2.x, and 1.3. It is an advanced virtualization management tool for the PowerVM and PowerKVM platforms. IBM Cloud Manager along with PowerVC Standard version provides support for the following capabilities:

- ▶ IBM Power Systems hosts that are managed by a Hardware Management Console (HMC)
- ▶ Storage area networks (SAN)
- ▶ Multiple Virtual I/O Server virtual machines on each host
- ▶ Multiple storage connectivity groups, which enable you to deploy images so that they have access to storage that is dedicated to a particular purpose

Limitations for PowerVC support from IBM Cloud Manager are as follows:

- ▶ Restarting and configuring drives are not supported for PowerVC 1.2.0.1 or 1.2.0.2.
- ▶ Image capture is not supported for any existing virtual machines that are added to PowerVC by using the Manage Existing function.

7.1.4 Deployment models

IBM Cloud Manager with OpenStack can be deployed in predefined configurations or topologies. The supported topologies are listed in Table 7-1.

Table 7-1 Supported cloud topologies

Topology	Description
Minimal	For product evaluation purposes. This topology is the simplest topology and does not require any customization.
Controller +n compute	For smaller production environments. This topology provides a single controller node, plus any number of compute nodes. You can configure this topology for your specific needs. For example, configure networking, resource scheduler, and other advanced customization.
HA controller +n compute	For larger test and production environments requiring high availability (HA) cloud controllers. This topology provides multiple HA controller nodes, plus any number of compute nodes. You can configure this topology for your specific needs.
Distributed database	For larger production environments. This topology is similar to the controller +n compute topology. However, it allows the IBM Cloud Manager with OpenStack database service to run on a separate compute node. It also supports advanced customization.
Multi-region	This topology is for larger test or production environments and can include multiple hypervisor environments. This topology is similar to the controller +n compute topology. However, you can separate hypervisors by region. Each region has its own controller, but shares the same Keystone and potentially IBM Cloud Manager Dashboard.

Minimum requirements: The minimal topology is not suitable for deploying to a PowerKVM or PowerVC environment. Be sure to check for Compute Hypervisor support when selecting a topology.

Figure 7-1 shows an example for *controller + n compute* topology option.

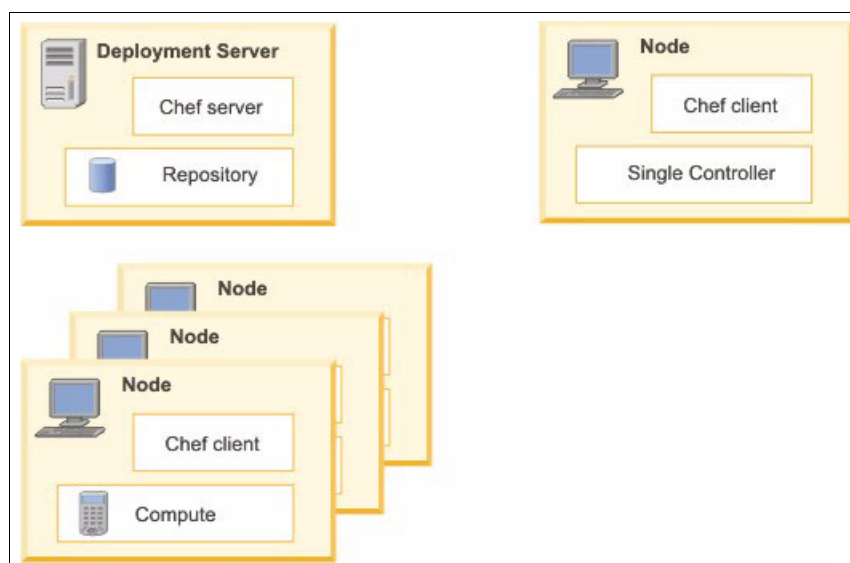


Figure 7-1 Controller + n compute topology

7.2 Identity

IBM Cloud Manager self-service portal supports role-based access control. It leverages Keystone or uses a Lightweight Directory Access Protocol (LDAP) service to enforce enterprise level identity management.

This section covers the following topics:

- ▶ 7.2.1, “Keystone and LDAP identities” on page 173
- ▶ 7.2.2, “Configuring LDAP” on page 174
- ▶ 7.2.3, “Projects, roles, and users” on page 175
- ▶ 7.2.4, “Changing default passwords” on page 176
- ▶ 7.2.5, “Changing the default administrator user account” on page 177

7.2.1 Keystone and LDAP identities

The IBM Cloud Manager self-service portal supports two user-authentication registries:

- ▶ OpenStack Keystone
- ▶ LDAP

By default, IBM Cloud Manager uses OpenStack Keystone database to store identity artifacts including credentials.

The use of an LDAP user registry is preferred for production environments to provide the highest level of security and identity management. The use of an LDAP user registry scales to hundreds or thousands of users and projects and supports TLS encryption.

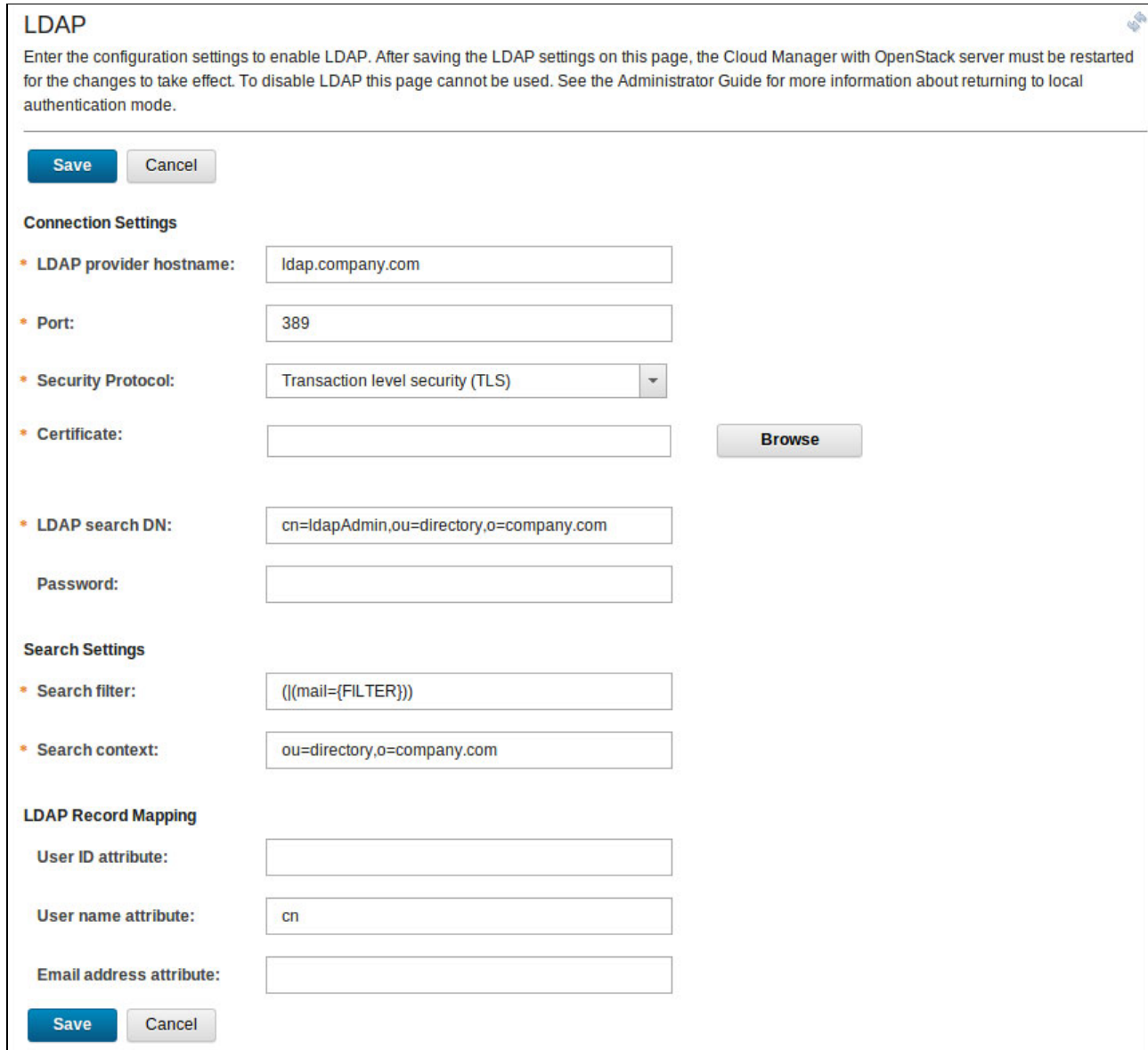
The following LDAP user registries are supported:

- ▶ IBM Tivoli® Directory Server Version 6.1
- ▶ OpenLDAP Version 2.4.x
- ▶ Microsoft Active Directory

7.2.2 Configuring LDAP

By default, the IBM Cloud Manager with OpenStack self-service portal uses the Keystone authentication mode. However, if you want the highest level of security, you should configure LDAP authentication.

The web interface is the primary means of configuring LDAP. Navigate to the LDAP configuration page in the Configuration tab and select **LDAP** in the navigation pane. Click **Edit** to enter the configuration settings. Figure 7-2 shows the LDAP configuration dialog box for IBM Cloud Manager with OpenStack self-service portal.

The image shows a web-based LDAP configuration dialog box. At the top, it has a title 'LDAP' and a paragraph of instructions: 'Enter the configuration settings to enable LDAP. After saving the LDAP settings on this page, the Cloud Manager with OpenStack server must be restarted for the changes to take effect. To disable LDAP this page cannot be used. See the Administrator Guide for more information about returning to local authentication mode.' Below this are 'Save' and 'Cancel' buttons. The form is divided into three sections: 'Connection Settings', 'Search Settings', and 'LDAP Record Mapping'. 'Connection Settings' includes fields for 'LDAP provider hostname' (ldap.company.com), 'Port' (389), 'Security Protocol' (Transaction level security (TLS)), 'Certificate' (with a 'Browse' button), 'LDAP search DN' (cn=ldapAdmin,ou=directory,o=company.com), and 'Password'. 'Search Settings' includes 'Search filter' (>((mail={FILTER}))) and 'Search context' (ou=directory,o=company.com). 'LDAP Record Mapping' includes 'User ID attribute', 'User name attribute' (cn), and 'Email address attribute'. At the bottom are another 'Save' and 'Cancel' buttons.

LDAP

Enter the configuration settings to enable LDAP. After saving the LDAP settings on this page, the Cloud Manager with OpenStack server must be restarted for the changes to take effect. To disable LDAP this page cannot be used. See the Administrator Guide for more information about returning to local authentication mode.

Save **Cancel**

Connection Settings

* **LDAP provider hostname:** ldap.company.com

* **Port:** 389

* **Security Protocol:** Transaction level security (TLS)

* **Certificate:** **Browse**

* **LDAP search DN:** cn=ldapAdmin,ou=directory,o=company.com

Password:

Search Settings

* **Search filter:** ((mail={FILTER}))

* **Search context:** ou=directory,o=company.com

LDAP Record Mapping

User ID attribute:

User name attribute: cn

Email address attribute:

Save **Cancel**

Figure 7-2 LDAP configuration dialog in self-service portal

See the IBM Cloud Manager with OpenStack documentation for detailed information about specific properties in the LDAP configuration page. The minimum required properties are LDAP provider hostname, Port, Security Protocol, Certificate, Search Filter, and Search context.

The remainder of the LDAP properties are optional, but be sure to enable TLS security on your LDAP server connection, otherwise passwords are transmitted over the network in clear text.

LDAP notes: The following mapping properties, if defined, must exist in the LDAP registry:

- ▶ User ID attribute
- ▶ User name attribute
- ▶ Email address attribute

For example, if you specify the User ID attribute to be `uid`, User name attribute to be `cn`, and Email address attribute to be `mail`, all three attributes (`uid`, `cn`, and `mail`) must exist for a user to be successfully created in the IBM Cloud Manager with OpenStack self-service portal.

After configuring LDAP in the self-service portal, restart the portal by using the following command so that LDAP becomes enabled:

```
service sce restart
```

You can also manually edit the `ldap.xml` and `authentication.properties` files. See the IBM Cloud Manager with OpenStack documentation for instructions on how to manually configure LDAP authentication.

User name case-sensitivity: If you want to enable the user name to be case-sensitive, edit the `ldap.xml` file after the initial configuration of LDAP by using the self-service portal interface.

For complete documentation about configuring IBM Cloud Manager to use an LDAP user registry, see the IBM Cloud Manager with OpenStack V4.3 documentation in the IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SST55W_4.3.0

7.2.3 Projects, roles, and users

Projects are used to define the users that have access to a set of images and instances. Only members of a project can view images and instances within a project. In many cases, projects correspond to a department or other organizational unit.

When an ID is added to a project, one of three memberships can be assigned to that identity.

Owner	A project owner has administrator authority to the project and its contents. The project owner primarily manages the contents of the project and has authority to the project and its contents.
User	A project user has the authority to use the project and the objects within the project. For example, a project user can deploy a virtual image to the project. A user can also view and potentially restore backup images of virtual machines that are created by other users, depending on how the project and roles were initially set up. The project user primarily handles their own deployments.
Viewer	A project viewer has authority only to view the project and the virtual images and instances that are contained in the project.

User and project management with OpenStack

Unlike other cloud types, OpenStack clouds provide native support for user and project management through the OpenStack Keystone component.

When you first connect to an OpenStack cloud, IBM Cloud Manager imports all the user accounts and projects that currently exist in OpenStack. All user roles and project membership are accepted and reflected in IBM Cloud Manager. After IBM Cloud Manager imports the initial OpenStack users and connects to an OpenStack cloud, IBM Cloud Manager enters transactional mode for user and project management. When in transactional mode, all operations that are performed in IBM Cloud Manager are also performed in OpenStack (for example, Keystone). If a user management operation (such as any of the operations that are described in this section) fails to complete successfully in IBM Cloud Manager, it does not occur in OpenStack. Likewise, if it fails in OpenStack it reverts in IBM Cloud Manager.

IBM Cloud Manager enters transactional mode for user and project operations while connected to OpenStack so that the user registries in both products are always synchronized. For this reason, when connected to an OpenStack cloud, performing user-related operations while the OpenStack cloud is down or unavailable is not possible.

To connect to OpenStack, IBM Cloud Manager uses a service user account and a default service tenant. Some installations of OpenStack have user accounts specific to OpenStack components (for example, Nova, Keystone, or Neutron). These and other service user accounts or service tenants in an OpenStack server that do not represent an actual user account or tenant can be added to the list of service users and service tenants so that they are ignored by IBM Cloud Manager. To make this change, add the service users and tenants to the comma-separated list in these properties:

- ▶ List of users in the `com.ibm.cfs.cloud.openstack.service.users` property,
- ▶ List of tenants in the `com.ibm.cfs.cloud.openstack.service.tenants` property, in the `openstack.properties` file.

Account management

Accounts are required when IBM Cloud Manager with OpenStack self-service billing is enabled. An account includes a balance, an owner, an account balance threshold, account members, and invoices. The account members are charged for the instances that they deploy.

Only administrators can create accounts, but a user can be assigned as an account owner.

7.2.4 Changing default passwords

The default system accounts and passwords used by IBM Cloud Manager with OpenStack are documented in the publicly available product literature, so an essential task is that the default password values are changed, either at initial topology deployment time, or after deployment.

Restriction: For a Minimal topology deployment, the passwords and secrets cannot be changed.

Changing passwords at deployment time

If you are deploying any other topology than Minimal, all passwords and secrets are obtained through encrypted *data bags*.

Follow the instructions in the “Customizing passwords and secrets” topic in the IBM Cloud Manager with OpenStack documentation to deploy with custom passwords. The basic steps are as follows:

1. Make a copy and update the example passwords and secrets file.
2. Set the passwords and secrets for your deployment.
3. Edit the data bags.
4. Update the changed data bags that are ready for deployment, by using the specified **knife** commands. The plain text passwords will be encrypted with the new secret key.
5. Delete the data bag directory because it is no longer needed.
6. Delete the passwords file because it is no longer needed.

Changing passwords after deployment

You can also change the passwords and secrets that were used in the deployment process after topology deployment.

Follow the instructions in the “Changing passwords and secrets” topics in the Modifying a Deployment for IBM Cloud Manager with OpenStack documentation to change the passwords and secrets used during the deployment.

7.2.5 Changing the default administrator user account

The default administrator account is created when IBM Cloud Manager self-service portal is installed. Use the following procedure to update the password of this administrator account and notification email account:

1. In the self-service portal, log in as the cloud administrator.
2. Select **Cloud Administrator** in the upper right title bar, and click **Show user preferences**.
3. In the Use Profile dialog, enter the administrator email address.
4. Select the **Send notifications about instances and other events** box.
5. Verify the **Timezone** and **Language** for the administrator.
6. To change the Cloud Administrator password, click **Change Password**.
7. Click **Update**.

7.3 Access

Network access to virtual machines is controlled by OpenStack Security Groups, which are the equivalents to access control lists (ACLs) on ingress and egress to the virtual machine networks.

This section covers the following topics:

- ▶ 7.3.1, “Access to provisioned virtual machines” on page 178
- ▶ 7.3.2, “Updating the default security policy” on page 178
- ▶ 7.3.3, “Generating and uploading SSH keys” on page 181
- ▶ 7.3.4, “Configuring SSL communication with self-service portal” on page 183
- ▶ 7.3.5, “Configuring SSL for OpenStack Dashboard” on page 184
- ▶ 7.3.6, “Network Time Protocol (NTP)” on page 185
- ▶ 7.3.7, “Session timeout and lockout” on page 185
- ▶ 7.3.8, “TCP/IP ports used by IBM Cloud Manager with OpenStack” on page 185

7.3.1 Access to provisioned virtual machines

Each tenant in an IBM Cloud Manager with OpenStack environment has a security group named *default*. The default security policies are as follows:

Ingress Allows all inbound network traffic from any virtual machine that is using this security group. Allows virtual machines that are deployed with this security group to receive data from each other. *To prevent or limit communication between deployed virtual machines, delete this rule and add a rule that is more restrictive.*

Egress Allows all outbound network traffic to be sent to all destinations (0.0.0.0). Does not restrict any outbound traffic from the virtual machines. *To limit outbound traffic from a deployed virtual machine, delete this rule and add a rule that is more restrictive.*

Restriction: It is possible to examine Security Groups using the Linux **iptables** command, but only modify the rules using the OpenStack dashboard or **neutron** command-line API.

7.3.2 Updating the default security policy

Updating security policies is done through the OpenStack dashboard. Select the security group you want to edit, for example select **default**, and click **Manage Rules**, as shown in Figure 7-3.

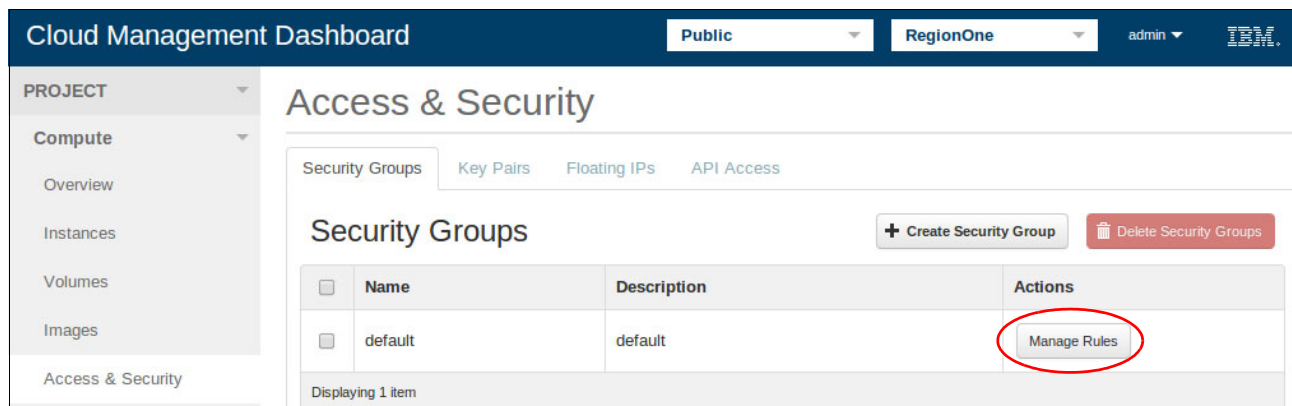


Figure 7-3 Security Groups panel in the OpenStack dashboard

The default security group ingress and egress rules are shown in Figure 7-4.

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote	Actions
<input type="checkbox"/>	Ingress	IPv4	Any	-	default	Delete Rule
<input type="checkbox"/>	Egress	IPv4	Any	-	0.0.0.0/0 (CIDR)	Delete Rule
<input type="checkbox"/>	Ingress	IPv6	Any	-	default	Delete Rule
<input type="checkbox"/>	Egress	IPv6	Any	-	:::0 (CIDR)	Delete Rule

Displaying 4 items

Figure 7-4 Ingress and Egress rules for the default security group

Add the rules you want. Predefined rules exist for commonly used protocols or applications, or you can define custom rules for TCP, UDP, ICMP, or other protocols. Figure 7-5 shows several predefined rules.

Add Rule

Rule: *
 Custom ICMP Rule
 Custom UDP Rule
 Custom ICMP Rule
 Other Protocol
 ALL ICMP
 ALL TCP
 ALL UDP
 DNS
 HTTP
 HTTPS
 IMAP
 IMAPS
 LDAP
 MS SQL
 MYSQL
 POP3
 POP3S
 RDP
 SMTP
 SMTPS
 SSH

0.0.0.0/0

Description:
 Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel Add

Figure 7-5 Adding rules to a security group using OpenStack dashboard

Consider restricting access to the minimal ports and protocols required. For example, instead of allowing ALL ICMP, consider allowing only ICMP echo (Type 8) and echo-reply (Type 0) to allow network pings, but not router redirects.

Depending on your network complexity, you might also allow the following ICMP types:

- ▶ 3 - Destination Unreachable
- ▶ 11 - Time exceeded
- ▶ 12 - Parameter problem

Figure 7-6 shows an example of the options available for a custom ICMP rule.

Add Rule

Rule: *

Custom ICMP Rule

Direction

Ingress

Type

Code

Remote: *

CIDR

CIDR

0.0.0.0/0

Description:

Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel Add

Figure 7-6 ICMP custom rule settings

Creating security group rules: You can also create security group rules using the CLI with the `neutron security-group-rule-create` command. See the OpenStack and IBM Cloud Manager with OpenStack documentation for the correct syntax. As opposed to a standard OpenStack environment, you also must specify a security group ID when using the CLI with IBM Cloud Manager with OpenStack.

For more information about configuring access and security for instances, see the following OpenStack web page.

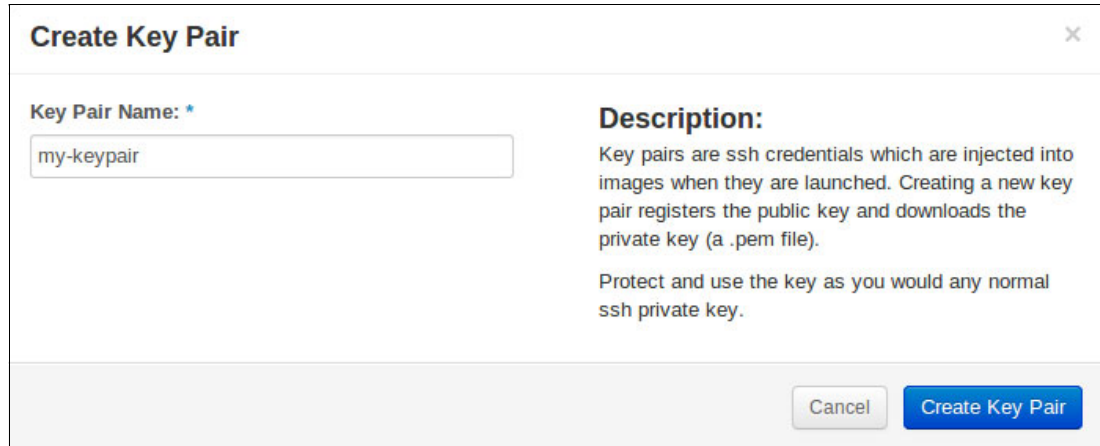
http://docs.openstack.org/user-guide/cli_nova_configure_access_security_for_instances.html#

7.3.3 Generating and uploading SSH keys

When a stored image is deployed to a hypervisor, user access is enabled by the copying of a predefined public key to the deployed virtual machine. The user then uses the corresponding private key to log in to the virtual machine, without the need to provide a password.

The following two methods provide user-specific SSH key pairs for the deployment:

- You can use the OpenStack dashboard to create a key pair. The private key is downloaded to your machine and the public key remains in OpenStack available for deployments. The Create Key Pair dialog box is shown in Figure 7-7.



Create Key Pair ✕

Key Pair Name: *

my-keypair

Description:

Key pairs are ssh credentials which are injected into images when they are launched. Creating a new key pair registers the public key and downloads the private key (a .pem file).

Protect and use the key as you would any normal ssh private key.

Cancel Create Key Pair

Figure 7-7 Creating a new key pair in the OpenStack dashboard

- You can upload a public key that you previously generated, as shown in Figure 7-8. You can copy and paste the public key into the text field as shown. Make sure that your copy and paste method does not introduce any spurious new lines or carriage returns.

Import Key Pair

Key Pair Name: *

my-imported-keys

Public Key: *

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDsN

Description:

Key Pairs are how you login to your instance after it is launched.

Choose a key pair name you will recognise and paste your SSH public key into the space provided.

SSH key pairs can be generated with the ssh-keygen command:

```
ssh-keygen -t rsa -f cloud.key
```

This generates a pair of keys: a key you keep private (cloud.key) and a public key (cloud.key.pub). Paste the contents of the public key file here.

After launching an instance, you login using the private key (the username might be different depending on the image you launched):

```
ssh -i cloud.key ubuntu@<instance_ip>
```

or:

```
ssh -i cloud.key ec2_user@<instance_ip>
```

Cancel Import Key Pair

Figure 7-8 Importing the public key of a pre-existing ssh key pair

Important: Regardless of the method used to store public keys in the OpenStack registry, you must be careful to protect your associated private key. If you lose this private key, you will not be able to log in to the deployed virtual machine. Conversely, if an unauthorized person has access to this private key, that person is able to access your virtual machines.

7.3.4 Configuring SSL communication with self-service portal

By default, the IBM Cloud Manager with OpenStack self-service portal includes a self-signed SSL certificate for HTTPS communication with the portal. This certificate is not created with the correct fully qualified domain name (FQDN) of the deployment server that you assigned in your environment. As a result, users receive a warning message when connecting to the self-service portal by using HTTPS.

Figure 7-9 shows a browser warning message for an untrusted SSL certificate. Also note that the connection (in this case, to the OpenStack Dashboard) is still encrypted, and in this case is using TLS 1.2 and the RC4_128 cipher.

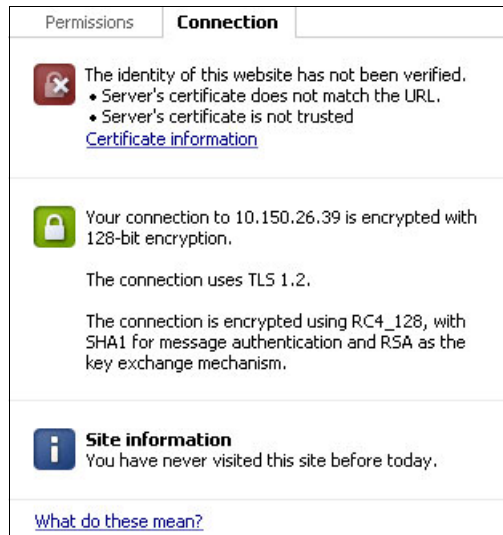


Figure 7-9 Warning message regarding untrusted SSL certificate

With untrusted certificates, a man-in-the-middle SSL certificate attack is more easily unnoticed. To secure your environment, replace this self-signed certificate with a certificate that is generated with the correct FQDN by a trusted internal or external certificate authority (CA).

In addition, the keystore password must be changed or another keystore must be used to contain this new certificate with a secure password. The new passwords are then used in the self-service portal's `server.properties` file. See Example 7-1 on page 184 for the properties contained in the `server.properties` file.

The properties to be modified to disable or enable HTTP and HTTPS services are indicated in the file comments.

Notes:

- Plain text passwords are replaced with encrypted representations when the self-service portal starts
- For the SSL protocol entry shown in Example 7-1, the following property is `SSL_TLS` if running on an IBM JRE; the property is `TLS` if running on a Sun or Oracle JRE.
`org.eclipse.equinox.http.jetty.ssl.protocol`

Example 7-1 Self-service portal server.properties file

```
# HTTP server port
org.osgi.service.http.port=18080

# Flag to enable/disable HTTP. If it is necessary for the protocol to be only SSL,
# set this flag to false.
org.eclipse.equinox.http.jetty.http.enabled=true

# Flag to enable/disable HTTPS
org.eclipse.equinox.http.jetty.https.enabled=true

# HTTPS port
org.eclipse.equinox.http.jetty.https.port=18443

# SSL password
org.eclipse.equinox.http.jetty.ssl.password=password

# Keystore password
org.eclipse.equinox.http.jetty.ssl.keypassword=password

# The full path location of the keystore
org.eclipse.equinox.http.jetty.ssl.keystore=home directory/.keystore

# The SSL protocol
org.eclipse.equinox.http.jetty.ssl.protocol=SSL_TLS
```

See the IBM Cloud Manager with OpenStack documentation for information about using **keytool** to generate a self-signed certificate, a certificate signing request (CSR), or a new keystore.

The following properties in the `server.properties` file must be updated with the new values for the keystore name and password:

- ▶ `org.eclipse.equinox.http.jetty.ssl.keystore`
- ▶ `org.eclipse.equinox.http.jetty.ssl.keypassword`

If the SSL certificate is generated with a password or passphrase, this is put into the `org.eclipse.equinox.http.jetty.ssl.password` property.

Default ports: By default, the IBM Cloud Manager with OpenStack self-service portal uses TCP port 18080 for unencrypted HTTP connections, and TCP port 18443 for encrypted HTTPS connections. You can change port numbers or disable HTTP connections in the `server.properties` file.

7.3.5 Configuring SSL for OpenStack Dashboard

By default, the OpenStack Dashboard redirects any non-secure HTTP requests on port 80 to the secure Apache HTTPS service on port 443. You can confirm this with your installation, and also change the SSL certificates being used in the `/etc/httpd/sites-enabled/openstack-dashboard` file.

7.3.6 Network Time Protocol (NTP)

Network Time Protocol (NTP) server attributes can be added to your environment deployment files, to synchronize time with the defined NTP servers across your topology. Update the following JavaScript Object Notation (JSON) attribute in your environment file:

ntp.servers Set to the NTP servers accessible to your deployment.

7.3.7 Session timeout and logout

The default self-service portal session timeout is after 30 minutes of inactivity. The default can be modified by setting the `com.ibm.cfs.client.idle.timeout` property in the `web.properties` file.

After three unsuccessful attempts to log in to the IBM Cloud Manager with OpenStack self-service portal, a user account is locked. Locked accounts are automatically re-enabled after 24 hours, or they can be unlocked by an administrator.

7.3.8 TCP/IP ports used by IBM Cloud Manager with OpenStack

The tables in this section contain lists of TCP/IP ports that are used by various components of IBM Cloud Manager with OpenStack. Consider these ports when you create firewall policies or create access control lists, either on hosts, or on network routers used for ingress or egress of TCP/IP traffic.

Table 7-2 lists the ports that are used by a single controller in IBM Cloud Manager with OpenStack.

Table 7-2 Port usage for IBM Cloud Manager with OpenStack single controller

Port	Service	Notes
22	sshd	SSH access that uses the customer network should be enabled.
67	DHCP server	
68	DHCP server	
80	openstack-dashboard-server	Provides access to the Horizon dashboard (non-secure).
443	openstack-dashboard-server	Provides HTTPS access to the Horizon dashboard (secure).
2224	pcsd	
3260	openstack-block-storage-volume-iscsi-port	
4369	rabbitmq-cluster-epmd	Provides access to ports used when a RabbitMQ cluster is configured.
5000	openstack-identity api	Identifies service public endpoint.
5405	pacemaker and corosync	
5671	openstack-messaging-server	
6080	openstack-compute-novnc	Computes VNC proxy for browsers.
7869	ego lim	Provides access to the load information manager in IBM Platform Resource Scheduler.

Port	Service	Notes
7870	ego vemkd	Provides access to the VEM kernel daemon in IBM Platform Resource Scheduler.
7871	ego pem	Provides access to the process execution manager in IBM Platform Resource Scheduler.
8000	openstack-orchestration-api-cfn	Orchestration AWK CloudFormation-compatible API.
8003	openstack-orchestration-api	Orchestration AWS CloudWatch-compatible API.
8004	openstack-orchestration-api	Orchestration endpoint.
8774	openstack-compute-api	Computes endpoint.
8776	openstack-block-storage-api	Blocks storage endpoint.
8777	openstack-telemetry-api	Ceilometer endpoint.
9191	openstack-image-registry	Image service registry.
9292	openstack-image-api	Image service API endpoint.
6385	openstack-ironic-api	
9696	openstack-network-api	Networking service API endpoint.
25672	rabbitmq-cluster-dist	Provides access to ports used when a RabbitMQ cluster is configured.
27017	openstack-database-nosql	DB2 NoSQL wire protocol listener for access to ceilodb2 database.
35357	openstack-identity-admin	Identifies service administrative endpoint.
50001	openstack-database-server	
50010 - 50017	db2-hadr	Provides access to ports used when DB2 HADR is configured.

Table 7-3 lists the TCP/IP ports that are used by IBM Cloud Manager with OpenStack self-service portal.

Table 7-3 Ports used by IBM Cloud Manager with OpenStack self-service portal

Port	Service	Notes
7777	sce	OSGi console, access from localhost only
8080	sce	IBM Cloud Manager - Self Service user interface
18443	sce	IBM Cloud Manager - Self Service user interface back-end

Table 7-4 lists the TCP/IP ports that are used by IBM Cloud Manager with OpenStack compute nodes.

Table 7-4 Ports used by IBM Cloud Manager with OpenStack compute nodes

Ports	Service	Notes
22	ssh	SSH port. This port must be accessible from the Chef server.
5900-5999	vnc-server	Only applicable to KVM/QEMU or PowerKVM compute nodes.

Table 7-5 lists the TCP/IP ports that are used by IBM Cloud Manager with OpenStack Chef server.

Table 7-5 Ports used by IBM Cloud Manager with OpenStack Chef server

Port	Notes
1480	The non-secure (HTTP) port for accessing the Chef server. The port number is configurable.
8443	The secure (HTTPS) port for accessing the IBM Cloud Manager - Deployer. The port number is configurable.
14443	The secure (HTTPS) port for accessing the Chef server. The port number is configurable.

7.4 Patch management

Fixes and updates for IBM Cloud Manager are provided on the IBM Fix Central website. You can use Fix Central to find and download the fixes that are recommended by IBM support for various products. IBM Cloud Manager with OpenStack is listed by selecting **Other Softer** under the Product Group:

<http://www.ibm.com/support/fixcentral>

Use these two steps to apply updates:

1. Update IBM Cloud Manager with OpenStack on the deployment server.
2. Update the topology on the client nodes.

Use the following steps to install a IBM Cloud Manager with OpenStack fix pack. The fix pack updates Chef cookbooks and other resources that are stored on the deployment server. It does not apply updates to the client nodes in your topology.

1. Copy the IBM Cloud Manager with OpenStack fix packs to a temporary directory, for example `/tmp/cmwo_fixpack`, on the deployment server.
2. Extract the fix pack files by using the following commands:

```
# cd /tmp/cmwo_fixpack
# tar -zvf cmwo_fixpack_4.1.0.1.tar.gz
```

3. Run the fix pack installation script:

```
# ./install_cmwo_fixpack.sh
```

The script output indicates whether the installation succeeded or failed and stores a copy of the fix pack log files in the `<product_dir>/version` directory, where `<product_dir>` is the directory where IBM Cloud Manager with OpenStack was installed. The default path is `/opt/ibm/cmwo`.

After you install the IBM Cloud Manager with OpenStack fix pack on the deployment server, you can deploy the fixes to your topology by using the **knife os manage update** tool. The tool uses the Chef server and client to update the client nodes, and can be used to update single nodes or all the nodes in your topology. Update the OpenStack controller and database nodes first, then update the compute nodes.

DB2 updates: DB2 updates are available separately to IBM Cloud Manager with OpenStack fix packs. See the IBM Cloud Manager with OpenStack or DB2 product documentation instructions for installing DB2 fix packs.

You can also subscribe to IBM My Notifications and receive customizable information containing important news and new or updated content, such as publications, hints and tips, technical notes, product flashes (alerts), downloads, and drivers. To stay up to date, subscribe to My Notifications at this web page:

<http://www.ibm.com/software/support/einfo.html>

Obtain operating system patches directly from the vendor, in this case Red Hat. For details about obtaining security updates from Red Hat, see this web page:

<https://access.redhat.com/security/updates>

7.5 Audit and logging

Log files are automatically saved by the IBM Cloud Manager with OpenStack self-service portal. You can configure the default number of log files that are saved and the types of messages that are logged.

By default, the self-service portal saves nine log files of 50 MB each. These defaults can be modified in the `logging.properties` file in the home directory. Logs are written to the `<HOME>/logs` directory.

Logging through syslog can be enabled by editing the following files:

- ▶ `/etc/nova/nova.conf`
- ▶ `/etc/keystone/keystone.conf`
- ▶ `/etc/glance/glance-api.conf`
- ▶ `/etc/glance-registry.conf`
- ▶ `/etc/cinder/cinder.conf`
- ▶ `/etc/iaasgateway/logging.conf`

To each file, add the lines shown in Example 7-2.

Example 7-2 Enabling syslog for OpenStack component logging

```
verbose = False
debug = False
use_syslog = True
syslog_log_facility = LOG_LOCAL0
```

The use of the syslog facility `LOG_LOCAL0` is arbitrary, and can be changed to another suitable facility. Restart OpenStack and syslog after changing the configuration files.

Configuring syslog logging allows the use of enterprise-level Security Information and Event Management (SIEM) tools such as IBM QRadar.

7.6 Image management

Images can be uploaded to the IBM Cloud Manager with OpenStack image library through the self-service portal or the OpenStack dashboard, or by using the `glance` command-line tools.

7.6.1 SSH host key entropy

Images in the image library should have their SSH host keys removed before being committed to the image library, or have a process in place to delete keys on image deployment. Duplication of SSH host keys across your environment, can introduce security risks, and potentially make factoring of private keys possible, through reduced randomness or entropy.

If SSH host keys, usually in `/etc/ssh/ssh_host*` files, are not present when SSH starts, it will generate new public and private host keys.

For more information about potential security implications of duplicate host keys, see the following website:

<http://factorable.net>

7.6.2 Image staging project

By default, IBM Cloud Manager with OpenStack scans the cloud for new images periodically. When IBM Cloud Manager with OpenStack finds a new image or instance, it places it in the Public project where all users have access to it.

To prevent the spread of malicious or poorly configured images, you can enable a staging project, where all newly discovered images or instances will be placed. Administrators can then review the images or instances before making them publicly accessible.

To enable the staging project, add the following line to the `deployment.properties` file, and then restart IBM Cloud Manager with OpenStack:

```
com.ibm.cfs.staging.project=Staging
```

7.7 REST API security

You can configure IBM Cloud Manager with OpenStack to require authentication when it calls to the IBM Cloud Manager with OpenStack REST APIs.

IBM Cloud Manager with OpenStack supports the following authentication methods:

- ▶ Basic HTTP authentication for a user login and REST API-based validation
- ▶ Encrypted token-based authentication for REST API calls

The basic strategy for using encrypted tokens is as follows:

1. HTTP/REST agents (browser or REST client) initially use the login authentication REST API to authenticate their user ID and password credentials.
2. The user ID and password are validated against the local or LDAP repository.
3. Upon successful login authentication, an encrypted token and its expiration are returned to the login response.
4. The agent can use (as an HTTP header cookie) the encrypted token for subsequent REST API calls to identify themselves until the token expires.
5. After the authentication token expires, the agent must use the login REST API again to validate their user ID and password. When complete, the agent obtains a new authentication token.

To enable authentication on calls to the REST APIs, set the `authentication.secure=true` property in the `authentication.properties` file. See the IBM Cloud Manager with OpenStack documentation for full details about the procedure.

7.8 Conclusion

IBM Cloud Manager with OpenStack is designed to be easy to install and easy to use. Be sure that after installation, you confirm or ensure that the following security considerations are implemented:

- ▶ Enable HTTPS on IBM Cloud Manager with OpenStack self-service portal and disable HTTP.
- ▶ Change the default passwords for all user and service accounts.
- ▶ Generate a new self-service portal SSL certificate using a trusted internal or external certificate authority.
- ▶ Generate a new keystore for self-service portal SSL certificate.



IBM Bluemix secure gateway

This chapter introduces the IBM Bluemix platform and provides an overview of how to connect on-premises Power infrastructure components through a safe secure environment named the *IBM Bluemix Secure Gateway*. This chapter also describes some of the other IBM Bluemix security features.

The following topics are covered in this chapter:

- ▶ 8.1, “IBM Bluemix overview” on page 192
- ▶ 8.2, “IBM Bluemix Secure Gateway” on page 195
- ▶ 8.3, “Other security options of IBM Bluemix” on page 202

8.1 IBM Bluemix overview

IBM Bluemix is the IBM open cloud platform based on platform as a service (PaaS) paradigm and built on Cloud Foundry open source technology. IBM Bluemix provides access for developers to IBM software for integration, security, transaction, and other key functions, and also access to software from IBM Business Partners.

IBM Bluemix is an abstraction layer of the IT infrastructure that hides most of the complexities that are associated with hosting and managing applications based on it. An application developer can focus on developing applications without the need to manage the infrastructure that is required to host it.

For both mobile and web applications, you can use the embedded services that are provided by IBM Bluemix. You can upload your web application to IBM Bluemix and indicate how many instances that you want to be run. After applications are deployed, you can scale the instances up or down when the use or load of the applications change by using a graphical user interface or an API. IBM Bluemix is an innovative cloud platform that fully links with the modern development and operations paradigm (DevOps).

The IBM Bluemix offering includes the following components:

- ▶ Application management
- ▶ Containers
- ▶ Virtual machines
- ▶ Data and analytics
- ▶ Services and API

You can use IBM Bluemix for developing applications in the most popular programming languages for platforms:

- ▶ iOS
- ▶ Android
- ▶ HTML with Java Script

For web applications, you can use languages such as these:

- ▶ Ruby
- ▶ PHP
- ▶ Java
- ▶ Go
- ▶ Python
- ▶ ASP.net
- ▶ Node.js

IBM Bluemix also provides middleware services for your applications to use. IBM Bluemix acts on the application's behalf when it provisions new service instances, and then binds those services to the application. Your application can perform its real job, leaving the management of the services to the infrastructure.

For more information about IBM Bluemix, see the IBM Bluemix website:

<http://www.ibm.com/cloud-computing/bluemix/>

8.1.1 How IBM Bluemix works

When you deploy an application to IBM Bluemix, you must configure IBM Bluemix with enough of the essential information to support the application:

- ▶ For a mobile application, IBM Bluemix contains an artifact that represents the mobile application's back-end, such as the services that are used by the mobile application to communicate with a server.
- ▶ For a web application, you must ensure that information about the proper run time and framework is communicated to IBM Bluemix, so that it can set up the proper execution environment to run the application.

As shown in Figure 8-1, each execution environment, including both mobile and web, is isolated from the execution environment of other applications. The execution environments are isolated even though these applications are on the same physical machine.

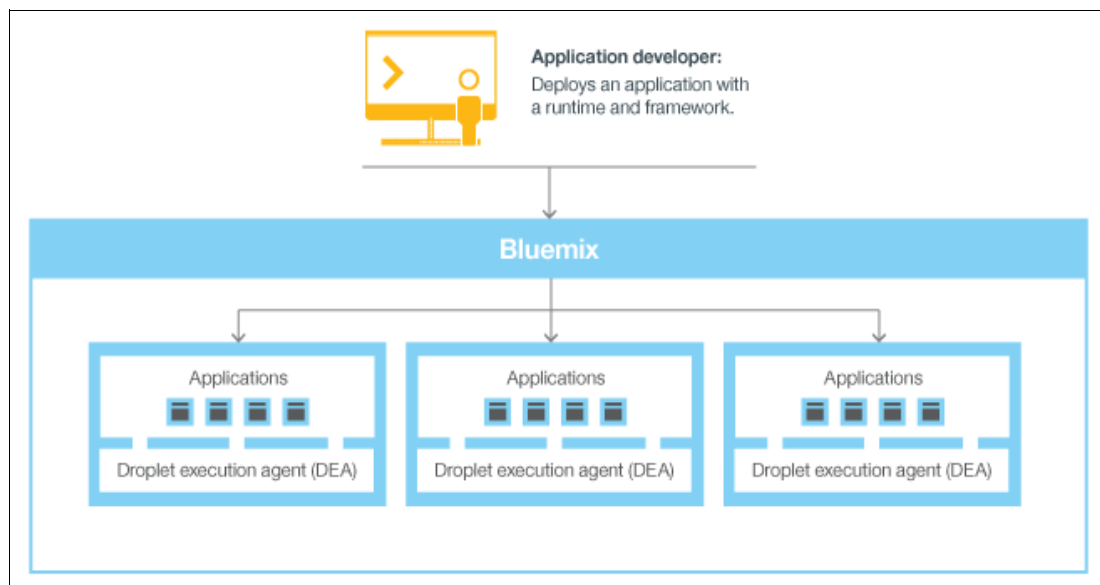


Figure 8-1 IBM Bluemix environment showing execution environments isolated from each other

When you create an application and deploy it to IBM Bluemix, the IBM Bluemix environment determines an appropriate virtual machine (VM) to which the application (or artifacts that the application represents) is sent:

- ▶ For a mobile application, a mobile back-end projection is created on IBM Bluemix. Any code for the mobile application running in the cloud eventually runs in the IBM Bluemix environment.
- ▶ For a web application, the code running in the cloud is the application that the developer deploys to IBM Bluemix.

The determination of which VM to use is based on several factors, including the load already on the machine and run times or frameworks supported by that VM. After a VM is chosen, an application manager on each VM installs the correct framework and run time for the application. Then, the application can be deployed into that framework. When the deployment is complete, the application are started.

In each VM, an application manager communicates with the rest of the IBM Bluemix infrastructure and manages the applications that are deployed to this VM. Each VM has containers to separate and protect applications.

In each container, IBM Bluemix installs the framework and run time that are required for each application. When the application is deployed, if it has a web interface (as for a Java web application) or other REST-based services (such as mobile services displayed publicly to the mobile application), users of the application can communicate with it by using normal HTTP requests.

Each application can have one or more URLs associated with it, but all of them must point to the IBM Bluemix endpoint. When a request comes in, IBM Bluemix examines the request, determines which application it is intended for, and then selects one of the instances of the application to receive the request.

8.1.2 IBM Bluemix management

For IBM Bluemix management, you can choose one of two options:

- ▶ IBM Bluemix Dashboard
- ▶ Cloud Foundry command-line interface

IBM Bluemix Dashboard

The IBM Bluemix Dashboard is a web-based management single point of control for your cloud routine tasks. With the IBM Bluemix dashboard, you can deploy and manage the application life cycle, performance, logs, and costs and perform tasks for user account management.

IBM Bluemix Cloud Foundry command-line interface

The Cloud Foundry **cf** command-line interface (CLI) is a powerful instrument for deploying and managing containers, applications, and service instances in an IBM Bluemix environment.

To set up the **cf** CLI and deploy applications, complete these steps:

1. Download and install the **cf** CLI from the following web page:

<https://github.com/cloudfoundry/cli/releases>

Note: The Cloud Foundry CLI is not supported by Cygwin.

2. Extract the package to a new directory to set up your environment:

```
cd new_directory
```

3. Connect to IBM Bluemix using the following **cf** command:

```
cf api https://api.ng.IBM Bluemix.net
```

4. Log in to IBM Bluemix:

```
cf login -u user_name -o org_name -s space_name
```

5. Deploy your application to IBM Bluemix using the following **cf push** command:

```
cf push appname
```

For more information about the **cf push** command, see the following web page:

https://www.ng.Bluemix.net/docs/starters/upload_app.html#upload_app_push

6. Access your application by entering the following URL into your web browser:

```
host.myIBM Bluemix.net
```


The **cf** CLI can be extended with extended command-line interface tools, such as these:

- ▶ IBM Containers for IBM Bluemix (ICE)
- ▶ IBM Bluemix Live Sync (BL)

You can also use plug-ins that help you to extend your IBM Bluemix command-line interface with more commands. To access the IBM Bluemix command-line interface plug-ins, see the IBM Bluemix CLI Plug-in Repository:

<http://plugins.ng.bluemix.net/ui/repository.html>

For more information about the IBM Bluemix documentation, see the following website:

<https://www.ng.bluemix.net/docs/>

8.2 IBM Bluemix Secure Gateway

The IBM Bluemix Secure Gateway service provides a secure way to access your on-premises or cloud data from your IBM Bluemix application through a secure passage.

As displayed in Figure 8-2, the service works by using a client to connect to your IBM Bluemix organization. Next, you add the service to your IBM Bluemix organization. Then, by using the Secure Gateway UI or REST API, you can begin creating your gateway by connecting to your client and creating a destination point to your on-premises or cloud data.

To increase security, you can add application-side TLS, which encrypts the data that travels from your application to the client. You can extend this security with client-side TLS, which encrypts the data from the client to the on-premises or cloud data.

When you complete your gateway configuration, you can monitor the behavior of your gateways and destinations in the IBM Bluemix Secure Gateway Dashboard. Using The IBM Bluemix Secure Gateway, you can securely connect the IBM Power server environment (especially the database environment) as the backend of front end applications running in IBM Bluemix.

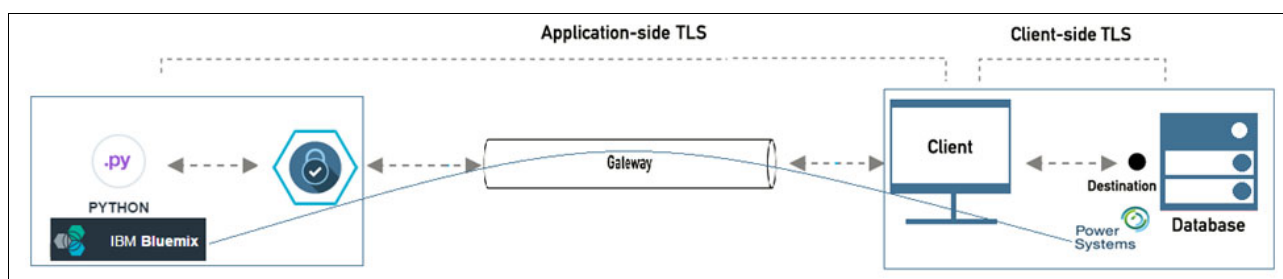


Figure 8-2 Using the IBM Bluemix Secure Gateway

Note: The IBM Bluemix Secure Gateway alone does not make possible the integration between the on-premises side and the public cloud side. It is just connects two sites and keep this connection secure. For integration between the client side and cloud side, you may use services such as the IBM Bluemix Cloud Integration Service or Data Power.

The IBM Bluemix Secure Gateway operates with the following terms:

Client	The process that establishes the on-premises or cloud side of the gateway and forwards requests to the destinations.
Gateway	The tunnel between your IBM Bluemix application and on-premises or cloud environment.
Destination	The point at which your on-premises data can be accessed.

8.2.1 IBM Bluemix Secure Gateway configuration

For IBM Bluemix Secure Gateway service, you need to make available the IBM Bluemix Secure Gateway connectors on both the client side and the cloud site. Use the following basic steps to set up the IBM Bluemix Secure Gateway connector on a IBM Bluemix site:

1. “Provision the service and bind it to your application” on page 196
2. “Prepare the client” on page 197
3. “Create the destination” on page 199

Provision the service and bind it to your application

To set up the IBM Bluemix Secure Gateway service use the following steps:

1. Create an application and bind it to Secure IBM Bluemix Secure Gateway service. These steps can be done by using the IBM Bluemix Dashboard.
2. After the application is created, run the IBM Bluemix Secure Gateway service and bind it to the application. To do this, go into your application and click **Add a service or API**.
3. Then select the IBM Bluemix Secure Gateway service, update the application and space-specific information, and click **Create**.

Note: To deploy a new instance of the application after it is bound to the service, click **RESTAGE**. See Figure 8-3.

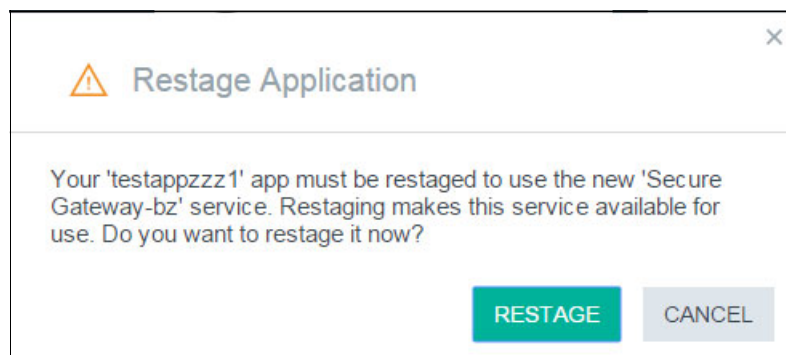


Figure 8-3 Deploying a new instance of an application bound to the Secure Gateway service

You can also use the following command to create and bind application to the service:

```
cf bind-service <appname> <service_instance>
```

After running these steps, the application will have the IBM Bluemix Secure Gateway service bound to it. Figure 8-4 shows the complete result of the binding.

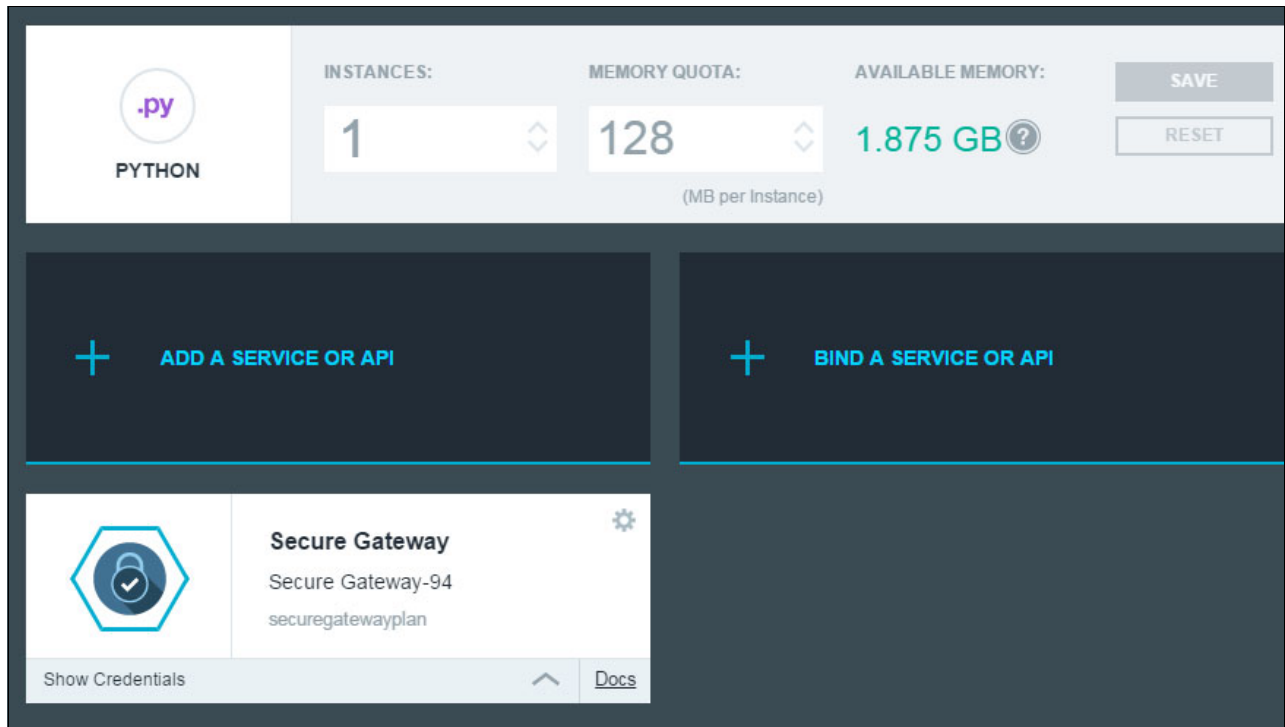


Figure 8-4 IBM Bluemix showing application bound to the Secure Gateway

Prepare the client

To successfully connect your IBM Bluemix Secure Gateway service to the client, first install and prepare the client applications. You can select the following options to install the client:

- ▶ Linux, Windows, or Mac OS X installer
- ▶ Docker image
- ▶ IBM DataPower® appliance

The Secure Gateway client provides client support for the following operating systems:

- ▶ Ubuntu Linux 14.04 Long Term Support (LTS) and greater
- ▶ Red Hat Linux 6.5 and greater
- ▶ SUSE Linux 11.0 and greater

The example shown here is for the Linux Ubuntu client setup. For more information about client installation, see the following website:

https://www.ng.bluemix.net/docs/services/SecureGateway/sg_021.html#sg_026

First, download the installer, as follows:

1. On the Secure Gateway Dashboard, click **Add Gateway** and provide a name for your gateway. Then, click **Connect it**.
2. On the Secure Gateway window that opens (Figure 8-5), select **IBM Installer** and then click the download link for your operating system.

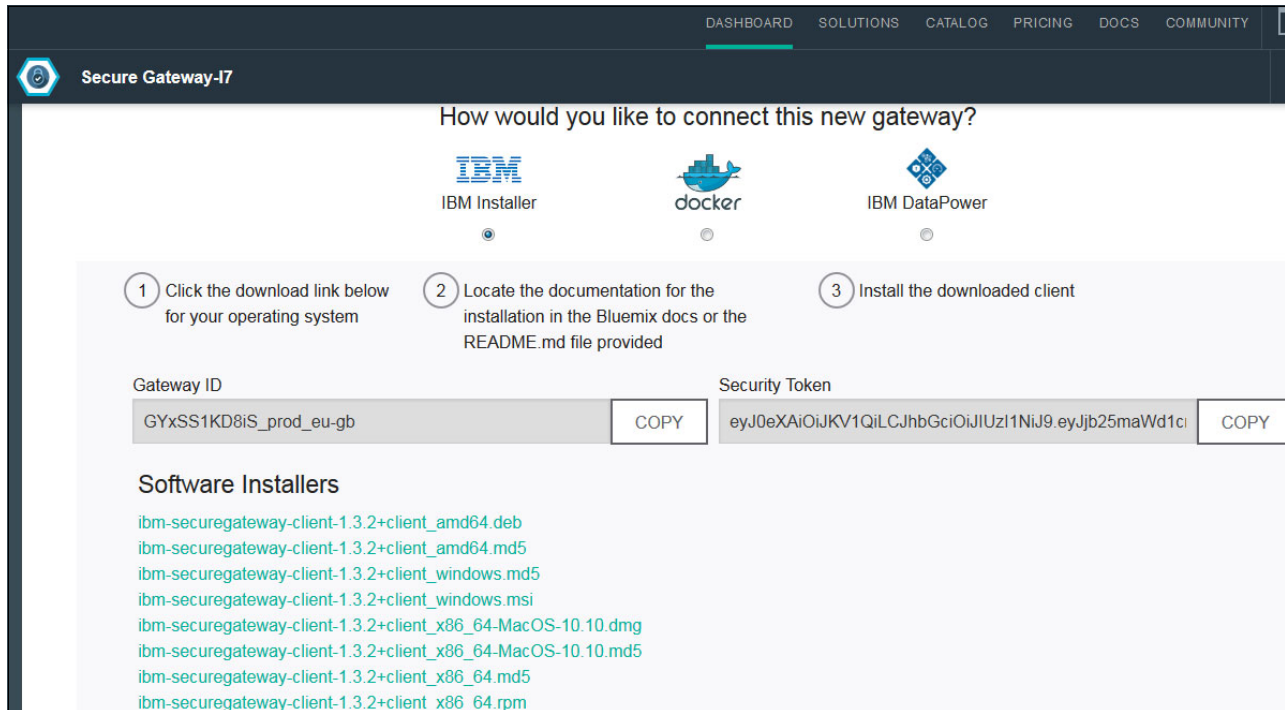


Figure 8-5 Selecting the IBM Bluemix Secure Gateway service installer for the client

3. After you download the client package, install it using a standard package manager:

```
sudo dpkg -i ibm-securegateway-client-1.3.0+client_amd64.deb
```
4. When the client installer starts, you are prompted for the following information:
 - **Auto-start process:** Yes or No
 - **Gateway ID:** The gateway ID is defined when you create a Secure Gateway service. If the client fails to connect, you can change your selection by editing the configuration file.
 - **Security token:** If the gateway ID is enabled to enforce security, you must provide the security token now. If the gateway ID does not require a security token, leave this blank.
 - **Logging level:** The default setting is info. Other valid values are trace, debug, or error.
5. To check the status of installation properties, run the following command:

```
sudo dpkg --status ibm-securegateway-client
```
6. The Secure Gateway client has a CLI and shell prompt for easy configuration and control. To start the Secure Gateway client, run the following command:

```
run secure-gateway-client
```

For descriptions of the Bluemix Secure Gateway commands, see Table 8-1.

Table 8-1 IBM Bluemix Secure Gateway commands

Command	Description
acl allow	Access control list allowed.
acl deny	Access control list denied.
acl file <file>	Access control list batch file.
no acl	Remove an access control list entry.
show acl	Show all access control list entries.
connect <gateway_id>	Connect to IBM Bluemix using the gateway ID that is provided.
gateway <hostname:port>	Manually select a specific gateway destination (advanced use only).
loglevel	Change the log level to ERROR, INFO, DEBUG, or TRACE.
logpath <file>	Direct logging to a specific file.
status	Print the status details of the tunnel and its connections.

Create the destination

A destination is a gateway connection to a specific on-premises resource. The host name and port number provide direct access to that resource from the cloud side.

To add destination from IBM Bluemix to on-premises site, complete the following steps:

1. From the IBM Bluemix Dashboard, find your IBM Bluemix Secure Gateway and open it. Click **Add destinations**.
2. The Create Destinations page opens (Figure 8-6 on page 200). Notice that the **Connect It** item is marked as complete.

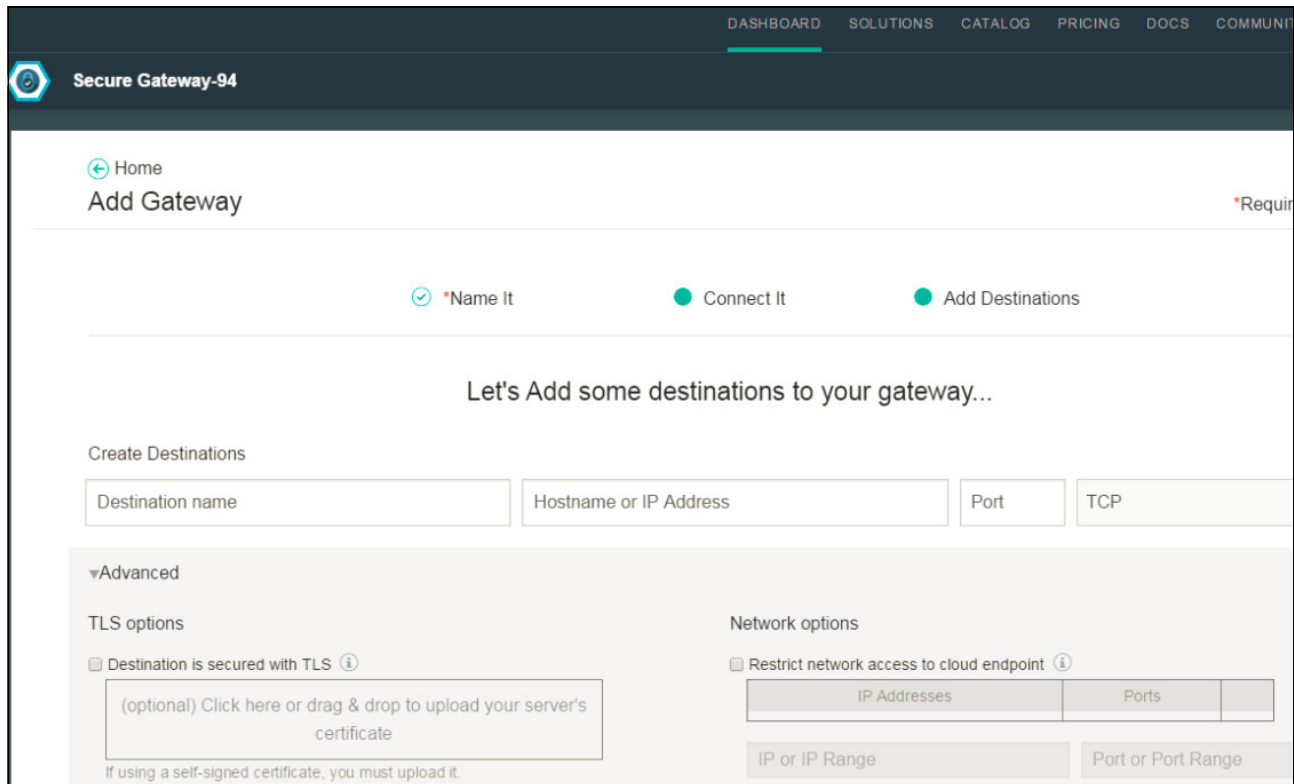


Figure 8-6 IBM Bluemix Secure Gateway dashboard, Create Destinations page

3. Enter the following information:

- Destination name
- Host name or IP address of your on-premises system
- Port number of your on-premises system
- Secure access selection

Specify how you want to secure access to the cloud or on-premises client. By default, TCP (Transmission Control Protocol) is selected.

Note: Without authentication. User application can communicate directly to the gateway without requiring any certificates.

- TLS: Server Side
TLS is enabled and the server provides a certificate to prove its authority.
- TLS: Mutual Auth
The server provides a set of certificates. However, you also need to upload your own certificate or select auto-generate to automatically create a self-signed certificate/key pair that you can download with the server certificate.
- HTTP
TCP connection where the host header is changed to match the on-premises host name.

- HTTPS

TLS connection where the host header is rewritten to match the on-premises host name. The TLS connection ends at the cloud server. To connect to a back-end HTTPS server, enable client-side TLS.

- HTTPS: Mutual Auth

TLS: Mutual Auth connection where the host header is changed to match the on-premises host name. The TLS connection ends after reaching the server. To connect to a backend HTTPS server, enable client-side TLS.

After completing these steps, you can connect to your destination from your IBM Bluemix application by using the cloud host and port number that is provided when you created the destination. For example, you can use a database connector mechanism. While connecting, you can monitor and gather gateway status and statistics.

8.2.2 IBM Bluemix Secure Gateway service status

The IBM Bluemix Secure Gateway provides status and usage statistics in interactive mode.

When the gateway is operating, you usually see messages logged indicating destination connections that are invoke to your on-premises resource. Logging is kept for these connections and can be viewed with the IBM Bluemix Dashboard or by using the Secure Gateway client in interactive mode.

Usage statistics can be collected by using **logstash** or **fluentd**.

After the client is running, use the **status** command to request information. Example 8-1 shows the result.

Example 8-1 Results of running the status command

```
[<Date & Time stamp>] [INFO] Connection #4 is being established to <IP address>:<Port no.>
[<Date & Time stamp>] [INFO] Connection #4 established to <IP address>:<Port no.>
[<Date & Time stamp>] [INFO] Connection #5 is being established to <IP address>:<Port no.>
[<Date & Time stamp>] [INFO] Connection #5 established to <IP address>:<Port no.>
Status for <gateway_id>
Overall BytesTX: 987BytesRX: 5942Total Connections: 6Active Connections: 6
ConnectionsHost: <IP address>Port: <Port no.>Bytes TX: 60Bytes RX: 81
Host: <IP address>Port: <Port no.>Bytes TX: 437Bytes RX: 2131
Host: <IP address>Port: <Port no.>Bytes TX: 124Bytes RX: 807
Host: <IP address>Port: <Port no.>Bytes TX: 121Bytes RX: 1058
Host: <IP address>Port: <Port no.>Bytes TX: 124Bytes RX: 807
Host: <IP address>Port: <Port no.>Bytes TX: 121Bytes RX: 1058
```

8.3 Other security options of IBM Bluemix

The following security service and APIs are available in the IBM Bluemix offering:

- ▶ Application Security Manager

The Application Security Manager provides a set of capabilities that enable organizations to take a strategic, risk-based approach to the application security problem.

- ▶ IBM AppScan® Dynamic Analyzer

The AppScan Dynamic Analyzer offers dynamic application security testing to the cloud, identifying security issues in web applications to help you keep them secure.

AppScan does not require any configuration. You only need to complete the URL field with the appropriate value (Figure 8-7).

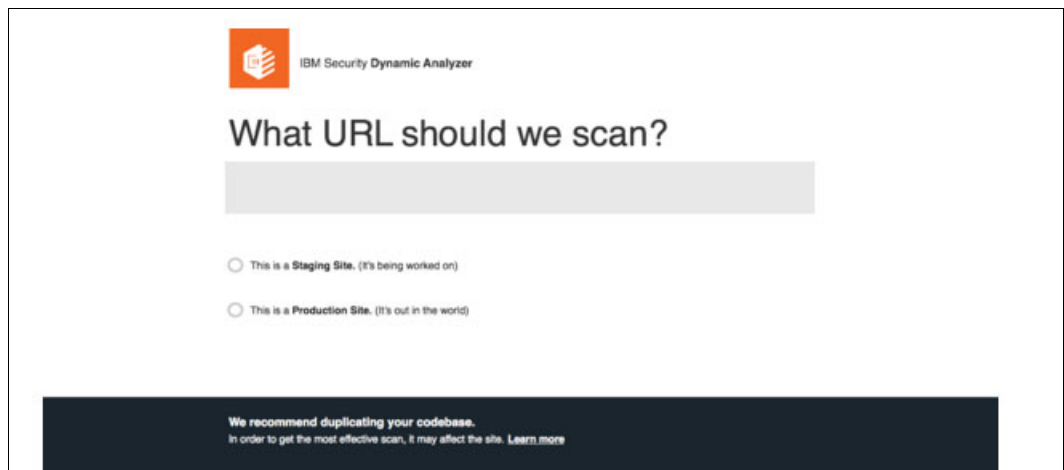
The screenshot shows the IBM Security Dynamic Analyzer interface. At the top left is the IBM logo, followed by the text "IBM Security Dynamic Analyzer". Below this is the heading "What URL should we scan?" followed by a large, empty text input field. Underneath the input field are two radio button options: "This is a Staging Site. (It's being worked on)" and "This is a Production Site. (It's out in the world)". At the bottom of the page, there is a dark blue banner with white text that reads: "We recommend duplicating your codebase. In order to get the most effective scan, it may affect the site. [Learn more](#)".

Figure 8-7 AppScan starting page

After the scan is finished, you can review a summary report page (Figure 8-8).



Figure 8-8 Results of the AppScan Dynamic Analyzer

► AppScan Mobile Analyzer

AppScan Mobile Analyzer is made for Android mobile applications. It identifies security issues in Android applications to help you keep them secure.

► Single Sign-On

Implement user authentication for web and mobile applications by using simple policy-based configurations.

► Static Analyzer

Static Analyzer brings the power of static application security testing to the cloud, scouring your code for unsafe data handling and API calls.

► aPersona Adaptive Security Manager

The aPersona ASM is a multitenant frictionless adaptive multi-factor authentication platform and service that protects web based transactions (including logins) from fraud and account takeover by unauthorized users.



Part 3

Appendixes

This part contains the following topics:

- ▶ Appendix A, “Troubleshooting SSL and TLS handshake” on page 207
- ▶ Appendix B, “VMware vRealize Automation for Power Systems” on page 215
- ▶ “Related publications” on page 219



Troubleshooting SSL and TLS handshake

You can use the AIX and Linux **tcpdump** and **wireshark** commands to examine the SSL handshake. Increasing security demands often result in mismatches in security capabilities between servers and endpoints when a secure connection is negotiated using Secure Sockets Layer (SSL) and Transport Layer Security (TLS).

An example might be where a user with an older web browser client is trying to connect to a secure web server that supports only a newer set of encryption protocols. For example TLS 1.2 might be mandated by a security standard.

In this case, a beneficial approach might be to examine the SSL or TLS encryption handshake to determine what encryption protocols and ciphers are supported by the client and server that are participating in the connection.

Network traffic can be captured for examination on the client or server by using **tcpdump**, and the resulting packet capture examined in a network protocol analyzer such as Wireshark.

The **tcpdump** command-line tool is available in AIX and Linux, and is ported to many other platforms. The **tcpdump** documentation and source code is available from this website:

<http://www.tcpdump.org>

The Wireshark graphical tool is available in most Linux distributions, and is also ported to other platforms. Wireshark documentation and source code is available from this website:

<http://www.wireshark.org>

Note: If Wireshark is installed on the host where you want to do the packet capture, you can collect the network stream directly from Wireshark.

The following topics are covered in this appendix:

- ▶ “Collecting network data by using tcpdump” on page 208
- ▶ “Examining packet captures with Wireshark” on page 208
- ▶ “Other tools” on page 213

Collecting network data by using tcpdump

The **tcpdump** command needs root or administrator access to put the network interface into promiscuous mode, and collect the network stream. With root privileges, run the following command to collect the required data:

```
tcpdump -i <interface> -w <filename> -s <snap-length> <filter>
```

In the command, *<interface>* is the NIC where you want to collect traffic from, *<filename>* is the file you want to write the data to, *<snap-length>* is the amount of data from each packet to capture, and *<filter>* is the tcpdump filter that you want to use to restrict data that is captured. Use of filter is optional, but can make processing easier if you can limit captured traffic to just the protocols, ports, or hosts that interested you.

Example A-1 shows capturing network traffic on interface eth0, writing to filename tcpdump.log, capturing the entire packet with snap-length 1500, and filtering for traffic only from a particular host with IP address 10.166.1.6. You can stop the packet capture by using Ctrl+C.

Example: A-1 Capturing traffic from a particular host using tcpdump

```
sh-4.1# tcpdump -i eth0 -w tcpdump.log -s 1500 host 10.166.1.6
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1500 bytes
^C4514 packets captured
4515 packets received by filter
0 packets dropped by kernel
```

After packet capture is complete, you can copy the tcpdump file to the host that has Wireshark available.

Examining packet captures with Wireshark

This section discusses the following topics:

- ▶ Introduction to SSL and TLS handshake
- ▶ Examining the SSL or TLS handshake

Introduction to SSL and TLS handshake

The official standards for SSL and TLS are available on the Internet Engineering Task Force (IETF) website, where the entire protocol definition is detailed.

Examples standards are as follows:

- ▶ SSL Protocol Version 3.0:
<https://tools.ietf.org/html/rfc6101>
- ▶ TLS Protocol Version 1.2:
<https://datatracker.ietf.org/doc/rfc5246>

Examining the SSL or TLS handshake

The basic process for a TLS handshake is as follows:

1. After TCP session establishment, the server can send a *Hello Request* at any time.
2. The client sends a *Client Hello* message. Of interest for this troubleshooting example, the *Client Hello* message contains a list of ciphers that the client supports (*cipher suite*). This is a list of the cryptographic options that are supported by the client, with the client's preference listed first.
3. The server sends a *Server Hello*, which contains the server's preferred cipher suite, and other protocol options, and the server's certificate data.
4. The client sends its certificate data in a *Client Key Exchange*.
5. The server completes the handshake protocol with an *Encrypted Handshake Message* and *Change Cipher Spec* message.
6. The client and server have now negotiated a cipher to use. A premaster secret key is then used to encrypt the remainder of the session, and little data of troubleshooting value can be derived from the packet trace.

Where a network protocol analyzer like Wireshark becomes valuable is to check the cipher suites that are supported by both client and server, because this might be a reason that a secure session cannot be initiated, or a session is not encrypted at a prescribed security standard.

The figures in this section show some relevant details from the example packet capture.

Figure A-1 shows a Wireshark packet detail of the Client Hello message, highlighting a particular cipher (TLS_RSA_WITH_RC4_128_SHA) from the cipher suite that has 23 entities.

No.	Time	Source	Destination	Protocol	Length	Info
39	20.138120	10.100.1.6	10.150.26.39	TCP	74	48099 > https [SYN] Seq=0 win=14600 L
40	20.138177	10.150.26.39	10.166.1.6	TCP	74	https > 48099 [SYN, ACK] Seq=0 Ack=1
41	20.331132	10.166.1.6	10.150.26.39	TCP	66	48099 > https [ACK] Seq=1 Ack=1 Win=1
42	20.562677	10.166.1.6	10.150.26.39	TLSv1.2	224	Client Hello
43	20.562729	10.150.26.39	10.166.1.6	TCP	66	https > 48099 [ACK] Seq=1 Ack=159 Win
44	20.563948	10.150.26.39	10.166.1.6	TLSv1.2	854	Server Hello. Certificate. Server Hel

Handshake Type: Client Hello (1)
Length: 149
Version: TLS 1.2 (0x0303)
▶ Random
Session ID Length: 0
Cipher Suites Length: 46
▼ Cipher Suites (23 suites)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
Cipher Suite: TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (0xc012)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_RC4_128_SHA (0xc007)
Cipher Suite: TLS_ECDHE_RSA_WITH_RC4_128_SHA (0xc011)
Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033)
Cipher Suite: TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x0032)
Cipher Suite: TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (0x0045)
Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x0038)
Cipher Suite: TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x0088)
Cipher Suite: TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x0016)
Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
Cipher Suite: TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (0x0041)
Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
Cipher Suite: TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x0084)
Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)

0010	00 02 76 18 40 00 3e 06 95 a5 0a a0 01 00 0a 90	..V.@.>.
0020	1a 27 bb e3 01 bb 30 b7 10 99 2f 14 80 6a 80 18	.'....0. ../.j..
0030	00 73 7b 74 00 00 01 01 08 0a 0c a3 66 be 05 54	.s{t.... .f..T
0040	ef 22 16 03 01 00 99 01 00 00 95 03 03 93 31 59	."..... ..1Y
0050	68 1b f2 84 e5 47 a1 60 8e 00 5d 1a 88 02 d3 a0	h....G.` ..]....
0060	4d ba 64 8f ce 9b 94 35 29 41 bc 8b 28 00 00 2e	M.d....5)A..(...
0070	c0 2b c0 2f c0 0a c0 09 c0 13 c0 14 c0 12 c0 07	././....
0080	c0 11 00 33 00 32 00 45 00 39 00 38 00 88 00 16	...3.2.E .9.8....
0090	00 2f 00 41 00 35 00 84 00 0a 00 05 00 04 01 00	./..A.5..

Figure A-1 Wireshark packet detail showing client cipher suite

Figure A-2 shows a wireshark packet detail of the Server Hello message with a cipher suite of only one entity (TLS_RSA_WITH_RC4_128_SHA).

No.	Time	Source	Destination	Protocol	Length	Info
40	20.138177	10.150.26.39	10.166.1.6	TCP	74	https > 48099 [SYN, ACK] Seq=0 Ack=1 W
41	20.331132	10.166.1.6	10.150.26.39	TCP	66	48099 > https [ACK] Seq=1 Ack=1 Win=14
42	20.562677	10.166.1.6	10.150.26.39	TLSv1.2	224	Client Hello
43	20.562729	10.150.26.39	10.166.1.6	TCP	66	https > 48099 [ACK] Seq=1 Ack=159 Win=
44	20.563948	10.150.26.39	10.166.1.6	TLSv1.2	854	Server Hello, Certificate, Server Hel
<div> <div>Version: TLS 1.2 (0x0303)</div> <div>Length: 53</div> <div> <div>▼ Handshake Protocol: Server Hello</div> <div>Handshake Type: Server Hello (2)</div> <div>Length: 49</div> <div>Version: TLS 1.2 (0x0303)</div> <div>▶ Random</div> <div>Session ID Length: 0</div> <div>Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)</div> <div>Compression Method: null (0)</div> <div>Extensions Length: 9</div> <div> <div>▶ Extension: renegotiation_info</div> <div>▶ Extension: SessionTicket TLS</div> </div> <div>▼ TLSv1.2 Record Layer: Handshake Protocol: Certificate</div> <div>Content Type: Handshake (22)</div> <div>Version: TLS 1.2 (0x0303)</div> <div>Length: 716</div> <div>▼ Handshake Protocol: Certificate</div> <div>Handshake Type: Certificate (11)</div> <div>Length: 712</div> <div>Certificates Length: 709</div> <div>▼ Certificates (709 bytes)</div> <div>Certificate Length: 706</div> <div>▶ Certificate (id-at-commonName=controller.example.com)</div> <div>▼ TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done</div> <div>Content Type: Handshake (22)</div> <div>Version: TLS 1.2 (0x0303)</div> <div>Length: 4</div> <div>▼ Handshake Protocol: Server Hello Done</div> </div> </div>						
0060	8c e2 0c ac 33 2a f2 03 7d 4e 0b 8f 4c 00 00 053*.. }N..L..				
0070	00 00 09 ff 01 00 01 00 00 23 00 00 16 03 03 02#.....				

Figure A-2 Wireshark packet detail showing server cipher suite

Figure A-3 shows a Wireshark packet detail of the certificate section of a Server Hello packet. You can check the basic details of the certificate, such as certificate host name (shown) and certificate validity dates.

No.	Time	Source	Destination	Protocol	Length	Info
40	20.138177	10.150.26.39	10.166.1.6	TCP	74	https > 48099 [SYN, ACK] Seq=0 Ack=1
41	20.331132	10.166.1.6	10.150.26.39	TCP	66	48099 > https [ACK] Seq=1 Ack=1 Win=1
42	20.562677	10.166.1.6	10.150.26.39	TLSv1.2	224	Client Hello
43	20.562729	10.150.26.39	10.166.1.6	TCP	66	https > 48099 [ACK] Seq=1 Ack=159 Win=
44	20.563948	10.150.26.39	10.166.1.6	TLSv1.2	854	Server Hello, Certificate, Server Hel
▶ Frame 44: 854 bytes on wire (6832 bits), 854 bytes captured (6832 bits)						
▶ Ethernet II, Src: Vmware 85:7c:5a (00:50:56:85:7c:5a), Dst: Ibm_e4:fb:00 (6c:ae:8b:e4:fb:00)						
▶ Internet Protocol Version 4, Src: 10.150.26.39 (10.150.26.39), Dst: 10.166.1.6 (10.166.1.6)						
▶ Transmission Control Protocol, Src Port: https (443), Dst Port: 48099 (48099), Seq: 1, Ack: 159, Len: 788						
▼ Secure Sockets Layer						
▶ TLSv1.2 Record Layer: Handshake Protocol: Server Hello						
▼ TLSv1.2 Record Layer: Handshake Protocol: Certificate						
Content Type: Handshake (22)						
Version: TLS 1.2 (0x0303)						
Length: 716						
▼ Handshake Protocol: Certificate						
Handshake Type: Certificate (11)						
Length: 712						
Certificates Length: 709						
▼ Certificates (709 bytes)						
Certificate Length: 706						
▼ Certificate (id-at-commonName=controller.example.com)						
▼ signedCertificate						
serialNumber : 0x00f3d01ffc44aa76f9						
▼ signature (shaWithRSAEncryption)						
Algorithm Id: 1.2.840.113549.1.1.5 (shaWithRSAEncryption)						
▼ issuer: rdnSequence (0)						
▶ rdnSequence: 1 item (id-at-commonName=controller.example.com)						
▼ validity						
▶ notBefore: utcTime (0)						
▶ notAfter: utcTime (0)						
▶ subject: rdnSequence (0)						
▶ subjectPublicKeyInfo						
▶ algorithmIdentifier (shaWithRSAEncryption)						
0080	cc 0b 00 02 c8 00 02 c5	00 02 c2 30 82 02 be 300...			
0090	82 01 a6 02 09 00 f3 d0	1f fc 44 aa 76 f9 30 0dD.v.0.			
00a0	06 09 2a 86 48 86 f7 0d	01 01 05 05 00 30 21 31	..*.H... ..0!1			
00b0	1f 30 1d 06 03 55 04 03	13 16 63 6f 6e 74 72 6f	.0...U.. ..contro			
00c0	6c 6c 65 72 2e 65 78 61	6d 70 6c 65 2e 63 6f 6d	ller.example.com			

Figure A-3 Server certificate details in wireshark

In this case, a protocol (TLS1.2) and a cipher (TLS_RSA_WITH_RC4_128_SHA) are both supported by client and server, and an encrypted session is established. This can be easily confirmed from the browser, after a secure connection is successfully negotiated, by clicking the padlock icon in the URL location bar.

Figure A-4 shows the server certificate detail as displayed in a Mozilla Firefox browser.

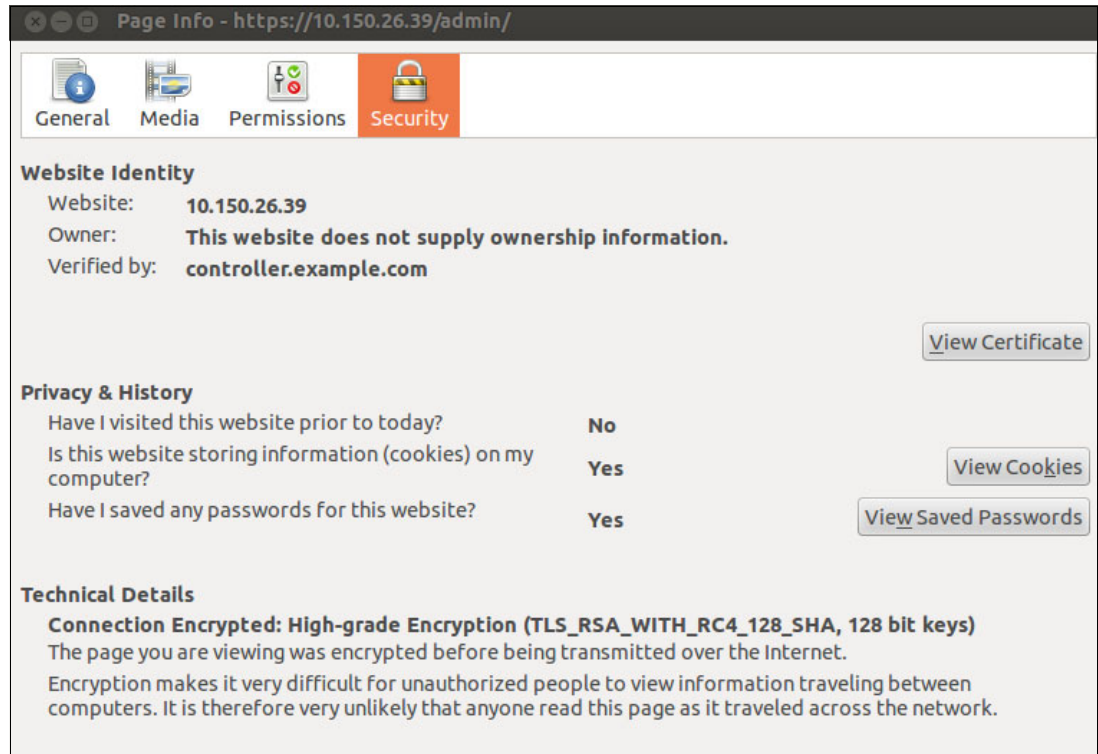


Figure A-4 Server certificate detail as displayed in a Firefox browser

Other tools

Server certificate information can also be gathered by command-line tools such as **openssl** with the **s_client** option, or **gnutls**. However, in cases where a secure SSL or TLS connection cannot be established using a particular client or application, basic troubleshooting with **tcpdump** and Wireshark can uncover the underlying problem.



VMware vRealize Automation for Power Systems

This appendix discusses the VMware vRealize Automation support for IBM Power Systems.

IBM and VMware announced a cooperative effort to offer a single tool for heterogeneous hybrid infrastructure cloud management. VMware vRealize Automation (vRA) makes calls to the OpenStack enabled APIs of PowerVC and IBM Cloud Manager with OpenStack to deliver this functionality. vRA manages PowerVC and IBM Cloud Manager with OpenStack as endpoints by using the northbound OpenStack APIs through REST.

Currently vRA supports the following guests on PowerVM:

- ▶ AIX
- ▶ Linux
- ▶ Linux guests on PowerKVM
- ▶ IBM i

Figure B-1 shows how the components work together and that vRA can also manage IBM z Systems™.

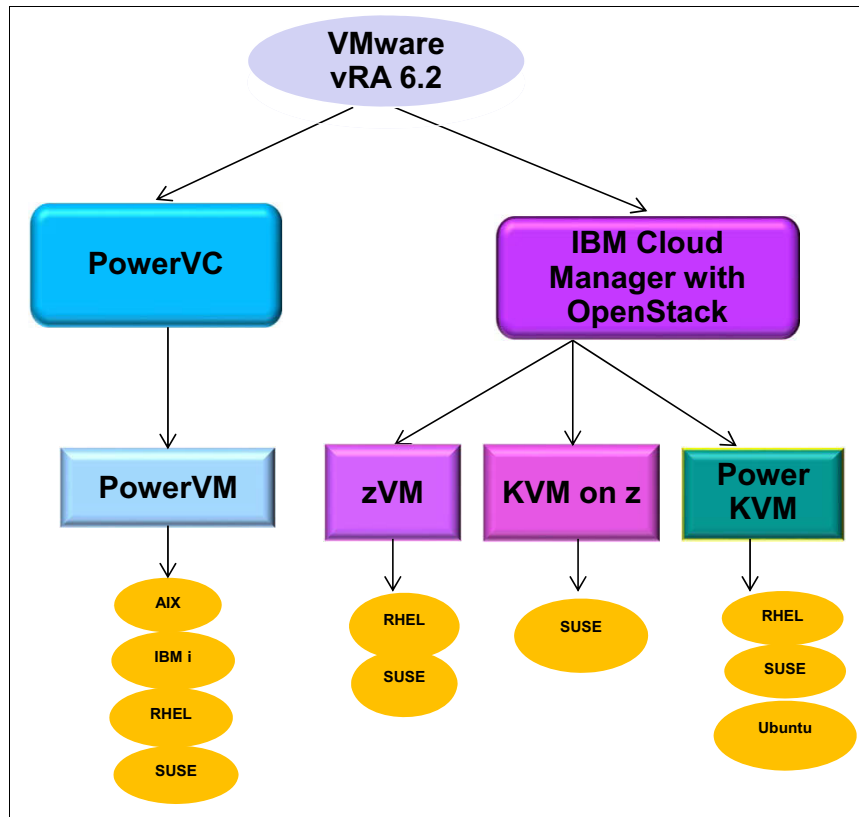


Figure B-1 How the vRA components stack together

Figure B-2 shows the heterogeneous hybrid cloud infrastructure.

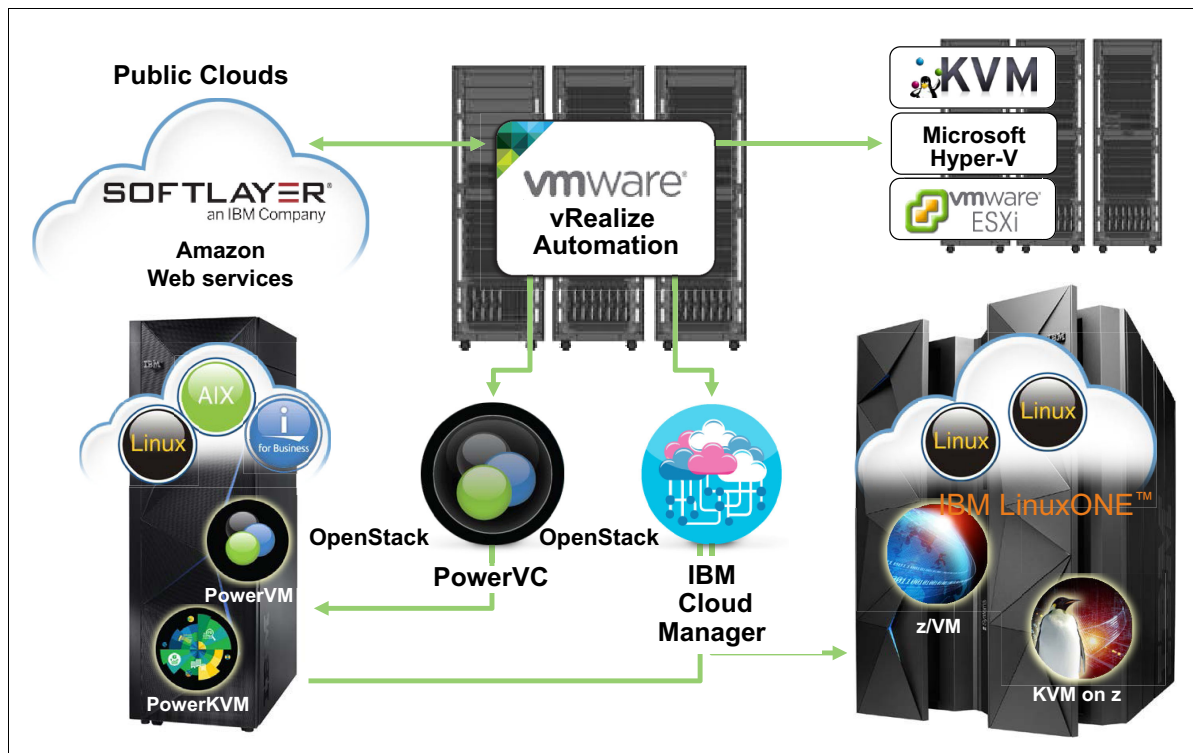


Figure B-2 Overview of the heterogeneous hybrid Cloud infrastructure

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.

- ▶ *IBM PowerKVM Configuration and Use*, SG24-8231
- ▶ *IBM Power Systems Hardware Management Console Version 8 Release 8.1.0 Enhancements*, SG24-8232
- ▶ *IBM Power Systems HMC Implementation and Usage Guide*, SG24-7491
- ▶ *IBM PowerVC Version 1.2 Introduction and Configuration*, SG24-8199
- ▶ *IBM PowerVM Virtualization Introduction and Configuration*, SG24-7940
- ▶ *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590
- ▶ *Integrated Virtualization Manager for IBM Power Systems Servers*, REDP-4061

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 resources are also relevant as further information sources:

- ▶ *Getting cloud computing right*, IBM Global Technology Services Thought Leadership White Paper:

<http://www.ibm.com/de/cloud/pdf/Gettingcloudcomputingright.pdf>

- ▶ *Under cloud cover*, IBM Center for Applied Insights, 2013. An IBM Study reveals that businesses using cloud computing for competitive advantage can generate double revenue and profit compared to their peers. At the following web page, click the **IBM survey** link:

<http://www.ibm.com/press/us/en/pressrelease/42304.wss>

- ▶ Federal Information Security Management Act (FISMA) of 2002

This federal law was enacted by the United States Congress. FISMA requires each federal agency to develop document and implement an agency wide program to provide information security for the information and information systems that support the operations and assets of the agency, including those provided or managed by another agency, contractor, or other source.

<http://www.dhs.gov/federal-information-security-management-act-fisma>

► National Institute of Standards and Technology (NIST)

NIST is responsible for producing many of the information security standards and guidelines that are used by United States Federal Agencies. In particular NIST is assigned many specific responsibilities, including the development of these items:

- Standards to be used by Federal agencies to categorize information and information systems based on the objectives of providing appropriate levels of information security according to a range of risk levels.
- Guidelines recommending the types of information and information systems to be included in each category; and
- Minimum information security requirements (management, operational, and technical security controls) for information and information systems in each such category.

See the NIST website:

<http://csrc.nist.gov>

► FedRAMP

The United States Federal Risk and Authorization Management Program (FedRAMP) is a government-wide program that provides a standardized approach to security assessment, authorization, and continuous monitoring for cloud service providers. This approach uses a “do once, use many times” framework that saves cost, time, and staff required to conduct redundant agency security assessments.

See the FedRAMP website:

<http://cloud.cio.gov/fedramp>

► Health Insurance Portability and Accountability Act (HIPAA) of 1996

HIPAA was enacted by the government to protect the privacy and security of individually identifiable health information. The HIPAA Security Rule sets national standards for the security of electronic protected health information.

The HIPAA mandates that all healthcare organizations effectively meet administrative, technical and physical safeguards to protect the privacy of patient information, and maintain data integrity for employees, customers, and shareholders.

See the Health Information Privacy web page:

<http://www.hhs.gov/ocr/privacy>

► Sarbanes-Oxley (SOX)

Following the corporate scandals of 2000, and a five trillion dollar dot.com crash, an overhaul of U.S. Securities Law resulted in the Sarbanes-Oxley Act of 2002 (often referred to as SOX), which has a complementary mission: “To protect investors by improving the accuracy and reliability of corporate disclosures made pursuant to the securities laws, and for other purposes.”

SOX was developed to address all the complexities of investor reporting, including individual accountability and integrity. Essentially, SOX requires corporations to make public the information that investors need for informed decisions. This means that IT departments, like other business entities, must constantly look for new and innovative ways to manage and report critical corporate information.

Search for Sarbanes-Oxley at the U.S. Securities and Exchange Commission (SEC) website:

<http://www.sec.gov>

► International Organization for Standardization (ISO)

The ISO and International Electrotechnical Commission (IEC) produce a family of information security standards, known as the ISO 27000 series.

Of particular relevance to cloud computing is the ISO 27001:2005 document, which deals with the specification of an information security management system.

See the ISO 27000 Directory website:

<http://www.27000.org>

► Common Criteria (CC)

CC is an international standard for computer security certification. A main role of CC is to provide the widest available mutual recognition of secure IT products. CC has replaced, or unified several pre-existing standards, into one widely recognized certification.

Common Criteria certifications scrutinize all security aspects of a product: design, source code, source code control, development process, and flaw remediation processes.

See the Common Criteria website:

<http://www.commoncriteriaportal.org>

► Payment Card Industry Data Security Standards (PCI-DSS)

The Payment Card Industry (PCI) Security Standards Council offers robust and comprehensive standards and supporting materials to enhance payment card data security. These materials include a framework of specifications, tools, measurements and support resources to help organizations ensure the safe handling of cardholder information at every step.

The Payment Card Industry Data Security Standard (PCI-DSS) applies to any organization that collects, stores, processes, or transmits credit card holder data, or interacts with a third-party company that does. Achieving and maintaining PCI-DSS compliance can be costly and time-consuming; non-compliance can result in extra fees or increased transaction charges. PCI-DSS is an internationally recognized standard with over 600 organizations participating.

See the PCI website:

<https://www.pcisecuritystandards.org>

► COBIT

Control Objectives for Information and Related Technology (COBIT) is a business framework for the governance and management of enterprise IT. COBIT was created by Information Systems Audit and Control Association (ISACA).

COBIT helps align information technology in support of business objectives by developing a framework for assessing and improving information technology strategy and policy within the organization. Corporate government initiatives often include identifying strategic investments in information technology and aligning information technology assets to business objectives.

See the ISACA COBIT website:

<http://www.isaca.org/cobit>

► BITS

BITS is the technology policy division of the Financial Services Roundtable (FSR). BITS addresses newly emerging threats and opportunities, particularly those related to cyber-security, fraud reduction and critical infrastructure protection in the financial services industry.

BITS was originally an acronym for Banking Industry Technology Secretariat, but that definition is no longer used.

See the BITS website:

<http://www.bits.org>

- Generally Accepted Privacy Principles

Generally Accepted Privacy Principles (GAPP) are privacy principles and criteria have been developed by American Institute of Certified Public Accountants (AICPA) and Chartered Professional Accountants (CPA) Canada to assist organizations in creating an effective privacy program that addresses their privacy risks and business opportunities

Search for GAPP at the following website:

<http://www.aicpa.org>

- OpenStack

Open source software for creating private and public clouds.

See the OpenStack website:

<http://www.openstack.org/>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



SG24-8242-01

ISBN 0738441422

Printed in U.S.A.

Get connected

