

Securing Your Cloud IBM Security for LinuxONE

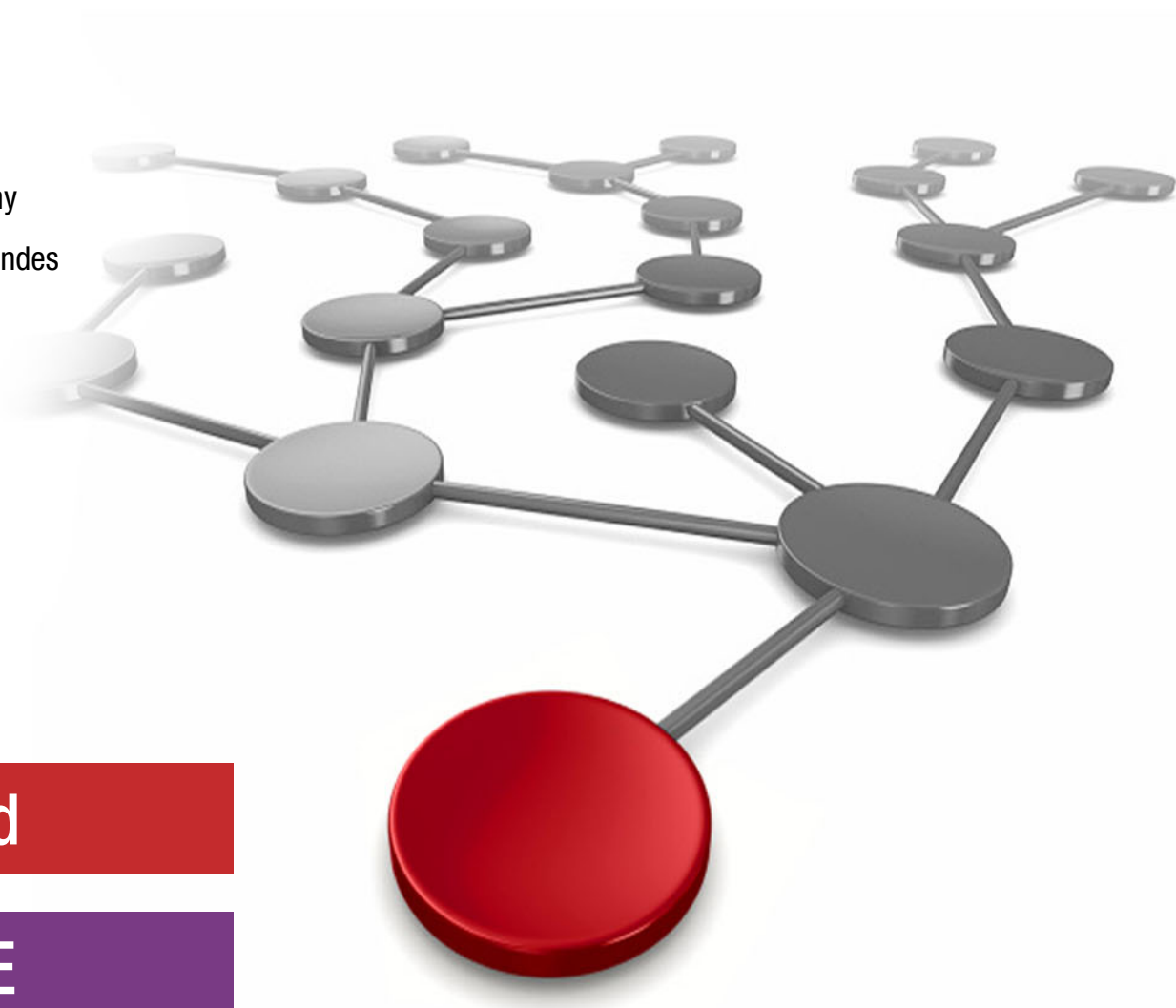
Edi Lopes Alves

Klaus Egeler

Karen Medhat Fahmy

Felipe Cardeneti Mendes

Maciej Olejniczak



 **Cloud**

LinuxONE



International Technical Support Organization

Securing Your Cloud: IBM Security for LinuxONE

July 2019

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

First Edition (July 2019)

This edition applies to Version 7, Release 1 of z/VM and the IBM Resource Access Control Facility Security Server for z/VM.

© Copyright International Business Machines Corporation 2019. All rights reserved.

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

Contents

Notices	vii
Trademarks	viii
Preface	ix
Authors	ix
Now you can become a published author, too!	x
Comments welcome	x
Stay connected to IBM Redbooks	xi
Chapter 1. IBM LinuxONE essentials	1
1.1 LinuxONE architecture and hardware	2
1.2 LinuxONE architecture	2
1.3 IBM LinuxONE servers	3
1.3.1 IBM LinuxONE Emperor II	3
1.3.2 IBM LinuxONE Rockhopper II	6
1.4 LinuxONE as a secure platform	9
1.4.1 The need for a secure platform	9
1.4.2 Security with LinuxONE	9
1.4.3 Using LinuxONE Security to create a secure cloud	11
1.4.4 IBM Hyper Protect Services overview	12
Chapter 2. Introduction to security on IBM LinuxONE	15
2.1 Why security matters	16
2.2 Hardware security features overview	16
2.3 Pervasive encryption	17
2.4 IBM LinuxONE cryptographic hardware features	18
2.4.1 CP Assist for Cryptographic Function	18
2.4.2 Crypto-Express6S	19
2.5 Benefits of hardware crypto	19
2.6 Using RACF to secure your cloud infrastructure	20
2.6.1 Principle of best matching profile	21
2.7 RACF DB organization and structure	22
2.7.1 Database definition to the system	22
2.7.2 Internal organization of RACF database specifying class options	22
Chapter 3. IBM z/VM hypervisor	25
3.1 Virtualization	26
3.1.1 Virtualization benefits	26
3.1.2 Hardware virtualization	27
3.2 z/VM hypervisor and LinuxONE servers	27
3.2.1 z/VM 7.1 overview	28
3.2.2 Single System Image overview	29
3.2.3 Security settings in an SSI cluster	31
3.2.4 Controlling the System Operator	32
3.2.5 System Configuration file	32
3.2.6 Addressing password security	35
3.2.7 Implementing CP LOGONBY	35
3.2.8 Role-based access controls and CP privilege classes	37
3.3 Device management	38

3.4	Securing the data	38
3.4.1	Securing your minidisks	39
3.4.2	Encrypting z/VM page volumes	39
3.4.3	Securing GUEST LANS and virtual switches	41
3.5	Securing your communication	42
3.5.1	Encrypting your communication	42
3.5.2	z/VM Cryptographic definitions	44
3.5.3	Checking the cryptographic card definitions in z/VM	48
3.6	z/VM connectivity	50
3.6.1	DEVICE and LINK statements	50
3.6.2	HiperSockets VSWITCH Bridge	51
3.6.3	Security considerations	52
3.7	Remote Spooling Communications Subsystem	52
Chapter 4.	IBM Resource Access Control Facility Security Server for IBM z/VM	55
4.1	RACF z/VM concepts	57
4.1.1	External security manager	57
4.1.2	Security policy	57
4.2	Activating and configuring RACF	59
4.2.1	Post-activation tasks	59
4.2.2	Building the RACF enabled CPLOAD MODULE	77
4.2.3	Updating the RACF database and options	80
4.2.4	Placing RACF into production	84
4.2.5	Using HCPRWAC	85
4.3	RACF management processes	88
4.3.1	DirMaint changes to work with RACF	88
4.3.2	RACF authorization concepts	90
4.3.3	Adding virtual machines and resources to the system and RACF database	90
4.3.4	Securing your minidisks with RACF	97
4.3.5	Securing guest LANs and virtual switches with RACF	99
4.3.6	Labeled security and mandatory access control	101
4.3.7	Backing up the RACF database	103
4.3.8	RACF recovery options	105
Chapter 5.	Security policy management on IBM z/VM	107
5.1	User ID management	108
5.1.1	Least privilege principle	108
5.1.2	RACF passwords and password phrases	114
5.1.3	Implementing RACF LOGONBY	123
5.2	Communication encryption	127
5.3	Single System Image Security	128
5.3.1	Overview	128
5.3.2	Equivalency identifiers	129
5.3.3	Relocation domains	129
5.3.4	RACF in an SSI cluster	130
5.4	Auditing	130
5.4.1	Auditing with journaling	131
5.4.2	Auditing with RACF	135
Chapter 6.	Securing a cloud in an IBM z/VM environment	157
6.1	Cloud on z/VM components	158
6.2	DirMaint	159
6.2.1	DirMaint controls	159
6.2.2	Delegating DirMaint authority	162

6.3 Systems Management API	167
6.3.1 SFS	167
6.3.2 Other SMAPI user IDs	168
6.3.3 VSMGUARD	169
6.3.4 SMAPI controls	170
6.3.5 Security aspects of SMAPI	170
6.4 z/VM Cloud Manager Appliance	174
6.4.1 Basic requirements and configuration options	175
6.4.2 OpenStack and xCAT Service Deployment Patterns	176
6.4.3 z/VM System Management Architecture	176
6.5 CMA Controller node	178
6.5.1 DMSSICNF COPY for the controller node	179
6.5.2 DMSSICMO COPY file for the controller node	180
6.6 CMA compute node	182
6.6.1 DMSSICNF COPY file for the compute node	182
6.6.2 DMSSICMO COPY file for the compute node	183
6.7 CMA installation	184
6.7.1 Initial set-up	186
6.7.2 Installing SMAPI 6.4 on your 7.1 system	186
6.7.3 Installing the CMA files on your z/VM 7.1 system	187
6.7.4 Restoring the CMA files	188
6.7.5 Configuring to use CMA 6.4 (Newton)	189
6.8 Securing your cloud components	190
6.8.1 Security considerations inherent in a cloud environment	191
6.8.2 Security tips for the cloud	193
Chapter 7. Securing IBM Cloud Private and Microservices on LinuxONE	195
7.1 Security in DevOps	196
7.2 Introduction to microservices	196
7.2.1 Microservice architecture	197
7.2.2 Service discovery	199
7.2.3 Securing your microservices application	200
7.3 Managing containers by using Kubernetes	202
7.3.1 Introduction to containers	202
7.3.2 Containers versus virtual machines	203
7.3.3 Container key points	204
7.3.4 Container orchestration	204
7.3.5 Kubernetes	206
7.3.6 Security in Kubernetes	208
7.4 Containers management at scale	213
7.4.1 IBM LinuxONE as the container platform	213
7.4.2 Deployment strategies	214
7.5 IBM Cloud Private overview	216
7.5.1 Key aspects	217
7.5.2 IBM Cloud Private architecture	217
7.5.3 IBM Cloud Private Security	218
7.5.4 IBM Cloud Private features	220
7.6 IBM Cloud Private on LinuxONE	223
7.6.1 Security levels for containerized applications on LinuxONE	223
7.6.2 IBM Secure Service Container	228
7.6.3 Deploying IBM Cloud Private on LinuxONE	230
7.6.4 IBM Cloud Private hands-on	233
7.6.5 Deploying a Node.js service on top of ICP and LinuxONE	234

7.7 IBM Cloud Automation Manager	239
7.7.1 Terraform	240
7.7.2 IBM Cloud Automation Manager on IBM Cloud Private	240
7.7.3 Security in IBM Cloud Automation Manager	242
Chapter 8. IBM z/VM and enterprise security	245
8.1 z/Secure	246
8.2 Lightweight Directory Access Protocol	246
8.2.1 LDAP on z/VM	247
8.2.2 Integration of z/VM LDAP into an enterprise directory	248
8.3 Linux on IBM LinuxONE security	249
8.3.1 Authentication	249
8.3.2 Access control	250
8.3.3 User management	251
8.3.4 Update management	251
8.3.5 Data	252
8.3.6 Audit	252
8.3.7 Cryptographic hardware	253
8.3.8 Firewall	254
Related publications	255
Other publications	255
Help from IBM	255

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.

BigFix®	IBM Cloud™	Redpaper™
DB2®	IBM LinuxONE™	Redbooks (logo)  ®
Db2®	IBM LinuxONE Emperor™	Storwize®
DirMaint™	IBM LinuxONE Emperor II™	System z®
ECKD™	IBM LinuxONE Rockhopper™	Terraform®
FICON®	IBM Spectrum™	Tivoli®
GDPS®	IBM Z®	WebSphere®
Geographically Dispersed Parallel Sysplex™	IBM z Systems®	z Systems®
Guardium®	Interconnect®	z/Architecture®
HiperSockets™	Parallel Sysplex®	z/OS®
IBM®	PR/SM™	z/VM®
IBM API Connect®	QRadar®	z/VSE®
IBM Blue®	RACF®	zSecure™
	Redbooks®	

The following terms are trademarks of other companies:

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

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

Microsoft, 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.

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

Preface

As workloads are being offloaded to IBM® LinuxONE based cloud environments, it is important to ensure that these workloads and environments are secure.

This IBM Redbooks® publication describes the necessary steps to secure your environment from the hardware level through all of the components that are involved in a LinuxONE cloud infrastructure that use Linux and IBM z/VM®.

The audience for this book is IT architects, IT Specialists, and those users who plan to use LinuxONE for their cloud environments.

Authors

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

Edi Lopes Alves is a Senior IT Specialist in Brazil working with IBM Z® and LinuxONE for the Global Technical Service team. She has more than 25 years of experience working as a z/VM and Linux on IBM Z specialist. Edi has IBM L2 IT Specialist certification, holds a Mathematics degree and a master's degree in e-Business from ESPM Sao Paulo. She currently supports z/VM and Linux on Z for the American Express and AIG accounts. She has supported the z/VM environment and cloud initiatives for Banco do Brasil and IBM Global Accounts (IGA) for several years by supporting IBM Green, IBM Blue® Harmony projects, and z/VM Field Test at Endicott Lab lpars. Working across international and diverse teams. Edi has co-authored several IBM Redbooks publications. Edi acquired several professional certifications, and has mentored several professionals at different levels of seniority to progress in their careers.

Klaus Egeler is an Senior IT Specialist in IBM's Research & Development Lab in Boeblingen, Germany. His area of expertise are IBM z/VSE®, z/VM and Linux on z and LinuxONE. Klaus has contributed to several z/VM-related and Linux-related IBM Redbooks and IBM Redpaper™ publications. He also is a presenter and instructor at workshops and customer events on a regular basis.

Karen Medhat Fahmy is an IBM L2 Certified working in IBM Egypt. She received her bachelor's degree with honors in Computer Engineering in 2012 and her MSc. degree in 2016 in the field of wireless sensor networks, security, and AI from the Faculty of Engineering, Cairo University. She joined IBM in 2013 and she is currently technical team leader in the cloud application innovation team and she has been developing and leading large-scale enterprise applications in several sectors. She has delivered different technical training sessions and courses as part of IBM Skills Academy for university Students across MEA. She has written several publications in the field of AI and IoT. She also received several technical and non-technical awards in IBM. Karen also has acquired several professional certifications, and contributed in developing and authoring IBM Cloud™ and IoT Certification Exams.

Felipe Cardeneti Mendes is an IT specialist with years of experience on distributed platforms and systems integration. His areas of expertise include Cloud, Docker containers, and virtualization across various platforms, including Intel x86, Power, LinuxONE, and IBM Z. Throughout his career, he worked on several integration projects and developed several successful solutions. He also frequently speaks at events and colleges to educate people about the latest Linux, Docker, and LinuxONE technology trends.

Maciej Olejniczak is an IBM Certified IT Expert. He has over 10 years of experience in IBM, 20 in IT. He is a cross-functional consultant in a collaborative environment, and is skilled in the design and delivery of integrated systems in various industries. He also works across international, diverse teams and is an IBM Redbooks Platinum Author. He excels at support delivery, problem solving, and managing critical situations. He is a member of IBM Academy of Technology and serves as a mentor to IBM employees and new hires. He also works within the IBM Security Team.

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

Lydia Parziale, Robert Haimowitz
IBM Redbooks, Poughkeepsie Center

Thomas Ambrosio, Bill Lamastro
IBM Competitive Project Office, Poughkeepsie, NY

William Romney, Scott Coyle, Steve Schultz
IBM Endicott Lab

Robert (Jay) Brenneman and Dulce Smith, POK

Guilherme da Silva Nogueira, Felipe Cardeneti Mendes, Sao Paulo
IBM Brazil

Now you can become a published author, too!

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

- ▶ Use the online **Contact us** review Redbooks form found at:
ibm.com/redbooks
- ▶ Send your comments in an email to:
redbooks@us.ibm.com

- ▶ Mail your comments to:
IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>



IBM LinuxONE essentials

Linux adoption grew dramatically over recent years, expanding from initial use by startups for web servers, into its use today for a vast range of enterprise computing workloads.

Digital is transforming the industry. To be competitive, enterprises must deliver trusted services to their clients while accelerating value. This need requires an open source platform that speeds your developers' creative genius and a highly secure cloud infrastructure that provides instantaneous data delivery any day, whether you have thousands or millions of simultaneous users.

LinuxONE is a Linux-only platform that supports customers who are interested in the use of the open source network that is combined with highly secure and highly scalable servers. This configuration makes it an efficient and cost-effective system for hosting various enterprise-grade Linux workloads.

LinuxONE is focused squarely at enterprise computing in the era of the cloud.

This chapter includes the following topics:

- ▶ 1.1, "LinuxONE architecture and hardware" on page 2
- ▶ 1.3, "IBM LinuxONE servers" on page 3
- ▶ 1.4, "LinuxONE as a secure platform" on page 9

1.1 LinuxONE architecture and hardware

LinuxONE can be configured to isolate or share data at different levels. Isolation can occur at the application, operating system, and hypervisor levels.

The option to fully encrypt data is available by using dedicated cryptographic processors. These cryptographic processors are in addition to the standard processing units that process the applications. This configuration helps gain valuable performance throughput by offloading the encryption tasks onto these specialized cryptographic processors.

The architecture must allow for changes in hardware and software to continue meeting rapidly changing needs. If the hardware and software components reach functional saturation, the architecture must expand (not change) to allow the hardware and software to grow and still be compatible with previous iterations and deployments. LinuxONE is founded on a solid and proven architecture.

This section describes the LinuxONE architecture, hardware, and its characteristics.

1.2 LinuxONE architecture

The LinuxONE family features the world's fastest commercially available processor. Built for speed, the platform supports simultaneous multithreading for Linux and Java workloads. It also helps deliver outstanding transaction processing and data serving performance.

IBM engineered LinuxONE as a highly scalable data-serving and transaction-processing platform that is vastly different from a standard x86-based server that runs Linux. In contrast, LinuxONE combines the enterprise qualities of the IBM hardware platforms with the openness of Linux and open source software.

LinuxONE is a mission-critical hardware platform with a unique shared memory and vertical scaling architecture. Dedicated Power and RAS cores in I/O channels and SAPs for I/O orchestration enable the platform to handle heavy I/O without compromising on latency and service millions of transactions per second.

Open-source Linux-Based software stack

LinuxONE's hardened Linux-based software stack can run most open source software packages, such as databases and data management (for example, MariaDB, PostgreSQL, MongoDB, and Apache Spark), virtualization platforms, containers (for example, IBM z/VM, KVM, and Docker), automation and orchestration software (for example, OpenStack, Puppet, Node.js, Juju, and Chef), and compute-intensive workloads, such as blockchain. In addition, LinuxONE Systems makes easy to build, model, deploy, and manage enterprise-grade scale-out clusters and scalable cloud architectures.

1.3 IBM LinuxONE servers

The first two LinuxONE products were named Emperor and Rockhopper. Emperor II and Rockhopper II, the second iteration of LinuxONE, were launched in 2017 and early 2018.

IBM LinuxONE™ features the following newest machines:

- ▶ IBM LinuxONE Emperor™ II

This machine features up to 170 processor cores, running at 5.2 GHz, up to 32 TB of RAM, and 640 dedicated I/O processors, all housed in a dual-frame. It supports tens of thousands of sessions and millions of containers. It can run 8,000 virtual servers and over 30 billion RESTful web interactions per day. This server is dedicated for Enterprise environments.

- ▶ IBM LinuxONE Rockhopper™ II

The same technology as Emperor II, but at a lower price. It is housed in an industry-standard, 19-inch rack. Rockhopper II is available with up to 8 TB of memory and 30 processor cores, running at 4.5 GHz. It supports hundreds of production and development virtual machines (VMs) in a single footprint.

For more information, see [this website](#).

1.3.1 IBM LinuxONE Emperor II

Emperor II is available with up to 170 configurable cores that use the world's fastest commercial processor for best performance and massive scaling.

The vertical scale allows Emperor II to scale up to 2 million Docker containers in a single system. It can serve up to 30 billion web data requests a day and host 20x larger databases without the added cost and latency of fragmenting data across server farms. Also, 640 processors are dedicated to I/O processing to increase I/O speeds and assure data integrity.

With 32 TB of real memory, Emperor II can open opportunities, such as in-memory data marts, large buffer pools for data access, and in-memory analytics. Advancements in the machine instruction set of the processor help accelerate analytic workloads by using the Vector Packed Decimal Facility, which allows packed decimal operations to be performed in registers rather than memory.

Java improvements, such as pause-less garbage collection, enables vertical scaling. The use of crypto-acceleration delivers more improvements in throughput per core, which provides a boost to Java processes that use cryptographic functions.

EAL 5+ isolation and cryptographic key protection

Emperor II has a EAL 5+ isolation and cryptographic key protection. EAL5+ is a regulatory certification for logical partitions (LPARs) that verifies the separation of partitions to improve security. Therefore, you can run many virtual servers concurrently, and use Emperor II's ability to isolate and protect each virtual server as though they were running on physically separated servers.

Emperor II provides isolation and cryptographic key protection by using a dedicated cryptographic coprocessor. The CP Assist for Cryptographic Function (CPACF) delivers cryptographic and hashing capabilities in support of clear-key operations. The Crypto-Express6S is used to create the fortified data perimeter by using the IBM LinuxONE protected key CPACF in which the keys that are used in the encryption process are not visible to the applications and operating system.

Galois Counter Mode

Galois Counter Mode (GCM) is a new feature of CPACF. The use of protected keys with GCM2 technology protects data without giving up performance, creating industry-leading secure Java performance by way of SSL.

Open source technology is driving the future and IBM is leading the charge with continued investment in the Linux network. Emperor II provides a unique platform for any Linux solution that requires high availability, security, or scalability, and supports a wealth of new open source products, such as Go, Python, Scala, Node.js, Docker, Spark, MongoDB, PostgreSQL, and MariaDB.

Emperor II allows clients to take advantage of transformative technologies, such as blockchain, gain cognitive insights by using Spark analytics, scale vertically with unmatched speed, and provide highly secure data serving capabilities.

Figure 1-1 shows the IBM LinuxONE Emperor II™ side view.



Figure 1-1 IBM LinuxONE Emperor II

For more information about the IBM LinuxONE Emperor II, see [this website](#).

LinuxONE Emperor II data sheet

The data sheet for Emperor II models is shown in Table 1-1.

Table 1-1 IBM LinuxONE Emperor II at a glance

IBM LinuxONE Emperor II features		
Emperor II Models	Cores: Min - Max	Memory: Min - Max
LM1	1 - 33	256 GB - 8 TB
LM2	1 - 69	256 GB - 16 TB
LM3	1 - 105	256 GB - 24 TB
LM4	1 - 141	256 GB - 32 TB
LM5	1 - 170	256 GB - 32 TB

IBM LinuxONE Emperor II features		
Cryptography		
Crypto-Express6S	Minimum 2 features; maximum 16 features	
Crypto-Express5S	Minimum 2 features; maximum 16 features	
Disk Connectivity		
IBM FICON® Express16S+/FICON Express16S/FICON Express8	Maximum: 320 ports	
NIC - Connectivity		
10 GbE RoCE Express2	Maximum 8; minimum recommended: 2	
OSA - Express6S	Maximum: 96 ports	
OSA - Express5S	Maximum: 96 ports	
High-speed “Virtual” LANS		
IBM HiperSockets™	Up to 32 connections	
Supported Linux distributors		
Red Hat	Red Hat Enterprise Linux (RHEL) 6 and 7	
SUSE	SUSE Linux Enterprise Server (SLES) 11 SP4, SLES 12 SP2, and SLES 15	
Canonical	Ubuntu 16.04 LTS and Ubuntu 18.04 LTS	
Supported hypervisors		
IBM z/VM	z/VM 6.4 (until the EOS) and z/VM 7.1 or higher	
KVM	KVM hypervisor, which is offered with the following Linux distributions: SLES-12 SP2 or higher, Ubuntu 16.04 or higher, and RHEL 7.5 or higher	
IBM partitioning technology	Up to 85 LPARs for secure workload isolation	
Typical physical weight of air-cooled configuration	Minimum configuration weight of new build LM1	Maximum configuration weight of new build LM5
With Internal Battery Feature (IBF)	LM1 1461 kg (3219 lb) with overhead cabling 1531 kg (3375 lb)	LM5 2705 kg (5961 lb) with overhead cabling 2775 kg (6117 lb)
Without Internal Battery Feature (IBF)	LM1 1258 kg (2772 lb) With overhead cabling 1328 kg (2928 lb)	LM5 2400 kg (5290 lb) With overhead cabling 2471 kg (5446 lb)
Product Dimensions (D x W x H) without overhead cabling	186.7 x 156.5 x 201.3 cm (73.5 x 61.6 x 79.3 in)	
Product Dimensions (D x W x H) with overhead cabling	186.7 x 184.7 x 215.3 cm (73.5 x 72.7 x 84.8 in)	
Airflow (Capacity of Exhaust)	6370 cubic meters per hour (3800 CFM)	

For more information about IBM tested and certified Linux environments, see this [web page](#).

1.3.2 IBM LinuxONE Rockhopper II

Rockhopper II delivers secure capabilities in a 19-inch frame with a lower cost of entry that can coexist with other platforms in any cloud data center. It is built on the strong foundation of the LinuxONE Emperor II platform.

Rockhopper II is housed in an industry-standard, 19-inch IBM-supplied rack. The design includes power distribution unit (PDU)-based power and redundant power, cooling, and power cords. These features allow you to install Rockhopper II within any data center with a server that is rated at ASHRAE A3. Up to 16U of available frame space can be used in the new 19-inch rack design.

Encryption and decryption features

Rockhopper II offers pervasive encryption. You can protect your assets by encrypting all data without compromising transactional throughput or response times. Rockhopper II includes dedicated encryption co-processors. The Central Processor Assist for Cryptographic Function (CPACF), which is standard on every core, supports pervasive encryption by providing hardware acceleration for encryption operations.

The Crypto-Express6S feature provides accelerated encryption, decryption, and tamper-sensing plus responding key management with a 2x SSL or TLS performance boost. Crypto-Express6S supports Accelerator for Secure Sockets Layer (SSL) transactions that are used to establish an encrypted link between a web server and a browser.

Keys never leave the secure coprocessor boundary unencrypted. This feature ensures that keys are not visible to the applications and operating system in clear text form.

Coupling facility (CF) encryption is a key piece of pervasive encryption that helps to protect CF data end-to-end by using encryption that is not apparent to applications. Although no encryption occurs on the CF, data is encrypted on a host in the sysplex by using CPACF on a per-workload, per-structure basis. The data that is written to the CF remains encrypted until it is read from the CF and decrypted by a host elsewhere in the sysplex.

Other features

IBM Secure Service Container uses LinuxONE's EAL5+ certification for vertical isolation of workloads and achieves horizontal isolation that separates the running application from the underlying Host environment. Secure Service Container is designed to offer the highest security level available for protected key management (FIPS 140-2 level 4).

For Rockhopper II, IBM Secure Service Container technology was enhanced for application deployment simplification. Clients and vendors can now run Docker container-based applications with which they can then integrate. Because Secure Service Container is now available as a service, clients can deploy their workloads. It is possible to scale up to 330,000 Docker containers in a single system.

The new FICON Express16S+ feature is designed to boost I/O rates and reduce single stream latency. The 16 Gb channel with a 3x start rate that is combined with a new 10 GbE RoCE Express2 adapter helps absorb large application and transaction spikes driven by unpredictable analytic and mobile workloads.

With double the available memory of the original Rockhopper, the 8 TB of real memory that is offered in Rockhopper II creates opportunities for in-memory data marts and large buffer pools for data access. The system is optimized for Java with a recent use of cryptographic acceleration and a pause-less garbage collection capability.

Rockhopper II provides a unique platform for any Linux solution that requires high availability, security features, or scalability. It also supports a wealth of open source products, such as Go, Python, Scala, Node.js, Docker, Spark, MongoDB, PostgreSQL, and MariaDB.

Rockhopper II allows clients to take advantage of transformative technologies, such as IBM Blockchain, gain cognitive insights with Spark analytics, scale vertically, provide highly secure data serving capabilities, and use the use of application programming interfaces (APIs).

Figure 1-2 shows the Rockhopper II from a side view.



Figure 1-2 IBM LinuxONE Rockhopper II

For more information about the IBM LinuxONE Rockhopper II, see [this website](#).

LinuxONE Rockhopper II data sheet

Table 1-2 provides some basic information about Rockhopper II models.

Table 1-2 IBM LinuxONE Rockhopper II at a glance

IBM LinuxONE Rockhopper II features		
Rockhopper II models	Cores: Min - Max	Memory: Min - Max
LR1 Max4	1 - 4	64 GB - 2 TB
LR2 Max12	1 - 12	64 GB - 4 TB
LR3 Max24	1 - 24	64 GB - 8 TB
LR4 Max30	1 - 30	64 GB - 8 TB
Cryptography		
Crypto-Express6S/Crypto Express5S	Minimum 2 features; maximum 16 features	
Disk connectivity		
FICON Express16S+/FICON Express16S / FICON Express8S	Maximum features (two ports per feature)	

IBM LinuxONE Rockhopper II features		
Max4	16	
Max12	32	
Max24, Max30	64	
NIC - Connectivity		
10 GbE RoCE Express2, 10 GbE RoCE Express	4 Maximum features (two ports per feature); minimum recommended is 2	
OSA-Express6S / OSA-Express5S / OSA-Express4S / 1000-BaseT	Maximum features (two ports per feature)	
Max4	16	
Max12	32	
Max24, Max30	48	
High Speed “Virtual” LANs		
HiperSockets	Up to 32 high-speed “virtual” local area networks	
Supported Linux distributors		
Red Hat	Red Hat Enterprise Linux (RHEL) 6 and 7	
SUSE	SUSE Linux Enterprise Server (SLES) 11 SP4, SLES 12 SP2, and SLES 15	
Canonical	Ubuntu 16.04 LTS and Ubuntu 18.04 LTS	
Supported Hypervisors		
IBM z/VM	z/VM 6.4 (until the EOS) and z/VM 7.1 or higher	
KVM	KVM hypervisor, which is offered with the following Linux distributions: SLES-12 SP2 or higher, Ubuntu 16.04 or higher, and RHEL 7.5 or higher	
IBM partitioning technology	Up to 40 LPARs for secure workload isolation	
Typical Physical Weight	Minimum configuration weight of new build 735 kg (1621 lb) Maximum configuration weight of new build 795 kg (1753 lb)	
Weight without side covers	Without overhead cabling 735 kg (1621 lb)	Overhead cabling adds approximately 5 kg (12 lbs) 740 kg (1633 lbs)
Weight with side covers adds approx. 42.7 lbs (19.4 kg)	Without overhead cabling 754 kg (1663 lb)	With overhead cabling adds approximately 5 kg (12 lbs) 760 kg (1675 lbs)
	Note: Optional seismic resistance hardware adds approximately 35 kg (78 lb)	
Product Dimensions (D x W x H) without side covers	Without overhead cabling: 107 x 60 x 201.5 cm (42.1 x 23.6 x 79.3 in)	With overhead cabling increases height 107 x 60 x 212.3 cm (4.3 inches 42.1 x 23.6 x 83.6 in)

IBM LinuxONE Rockhopper II features		
Product Dimensions (D x W x H) with side covers	Without overhead cabling 120.4 x 62.4 x 202 cm (47.4 x 24.6 x 79.5 in)	With overhead cabling increases height 120.4 x 62.4 x 212.8 cm (47.4 x 24.6 x 83.8 in)
Airflow (Capacity of Exhaust)	2000 cubic meters per hour (1200 CFM)	

For more information, see the IBM Systems Data Sheet [IBM LinuxONE Rockhopper II](#).

1.4 LinuxONE as a secure platform

Security must be at the center of any IT platform. If critical business data is compromised or customer data is leaked, your business's reputation can be damaged, and you can face regulatory and legal consequences. Likewise, if corporate data is exposed, you risk the chance of losing significant intellectual property.

When you are considering an infrastructure platform, you must understand the security features that are inherent in the platform in the cloud and on premises. In this section, we discuss how the LinuxONE system incorporates a high level of security.

1.4.1 The need for a secure platform

Initially, corporate management assumed that regulatory compliance and audits are enough to protect your company's data. However, many security risks come from third-party malicious attacks. Management now understands that with the advent of cloud computing, many of the risks can be out of their direct control.

Businesses are concerned about cybersecurity threats to the information that is the lifeblood of their relationships with their customers and partners. More often, data is in a multicloud environment, and applications are designed to manage data and provide collaboration between customers and partners.

1.4.2 Security with LinuxONE

Security is built in at the lowest level of the platform for LinuxONE for the hardware and software. The most important technologies for ensuring this level of protection are pervasive encryption, Hardware Security Module (HSM), and IBM Secure Service Container, as described in "Other features" on page 6.

The idea of encrypting all your data is new. Because of the overhead of software encryption, businesses in the past were forced to choose which data to encrypt. This decision left the remaining data at risk. In nearly every online interaction, data is left unencrypted at some point in the process. When data is left unencrypted, this point provides an opportunity to steal data.

Pervasive encryption

Pervasive encryption can encrypt data that is at rest and in flight. This type of encryption does not require application changes. This approach enables companies to encrypt all their data by default with little compute overhead.

One of the benefits of the LinuxONE system is the extent of the security services. Because of the architecture of LinuxONE, security is pre-integrated at every level of the hardware and software stack. LinuxONE-based security is designed to encrypt data in bulk. Therefore, it is possible to encrypt all the data that is associated with an application or a database at once.

Providing encryption of everything and at every level is in stark contrast to the way encryption is typically approached. Most companies encrypt only a small amount of data. However, when all the data is encrypted, even if it is made available to people outside of your organization, it is meaningless without the encryption key.

Traditionally, encrypting all of your data required a large amount of compute and time overhead. However, the LinuxONE platform includes dedicated hardware that is tuned for encryption.

The one-chip encryption co-processor is on every compute chip next to the main processor and can encrypt up to 13 GB of data per second per core.

This level of protection is achieved through hardware accelerated encryption of data, which is delivered with near-zero overhead by the on-chip Central Processor Assist for Cryptographic Function (CPACF) and the new dedicated Crypto-Express6S adapter. The availability of this level of encryption makes it easier for applications to meet regulations such as HIPAA and PCI DSS.

Figure 1-3 shows the pervasive encryption from a virtualization perspective.

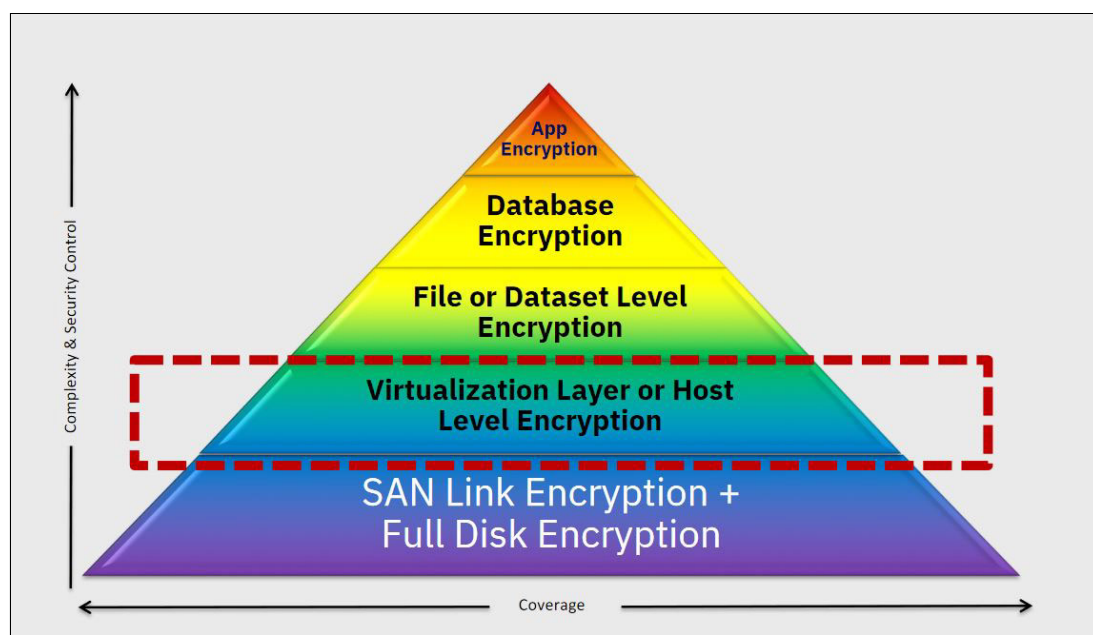


Figure 1-3 IBM LinuxONE pervasive encryption

Hardware Security Module

LinuxONE can also include CryptoExpress adapters, which support high-speed encryption and an HSM. Security is further promoted by protecting cryptographic keys by using an HSM.

These cryptographic co-processors are protected within a tamper-responsive environment that destroys encryption keys if it senses an attack.

Protected key encryption is processed in the CPACF for high speeds and stored in an HSM. This key encryption enables fast encrypting and decrypting of complete disks (volumes) or selected partitions. LPAR isolation, which is standard on all LinuxONE processors for generations, virtually eliminates east-west, north-south security breaches, and their damaging effect both financially and to an organization's credibility.

IBM Secure Service Container

The IBM Secure Service Container for IBM Cloud Private (ICP) is a solution that hosts container-based applications for hybrid and private cloud workloads on IBM LinuxONE. This secure computing environment for microservices-based applications can be deployed without code changes to use the security capabilities and provides the following benefits:

- ▶ Tamper protection during installation and start time to protect against malware attacks
- ▶ Restricted administrator access to help prevent the misuse of privileged user credentials for cloud and on premises environments
- ▶ Automatic pervasive encryption of data that is in flight and at rest

IBM Secure Service Container adds other security capabilities for Docker and other container environments. Linux provides a comprehensive set of security technologies, including firewalls, VPNs, auditing tools to support regulatory compliance, and SELinux, which is a kernel-based security system.

IBM Secure Service Container technology builds on the workload isolation of the firmware that is based LPARs and is unique to IBM LinuxONE. It was used in the IBM Cloud on LinuxONE to provide the advanced security of IBM Blockchain Platform and is now extended for generic container-based applications through IBM Secure Service Container for ICP.

For more information, see CHAPTER 7 (ICP).

1.4.3 Using LinuxONE Security to create a secure cloud

Organizations can apply LinuxONE security services to their computing environments by using different methods. These services can be used as part of a private cloud or as cloud services that are hosted in ICP. For example, a client can implement an on-premises LinuxONE machine to build a private cloud, or it can gain access to the secure services by provisioning a LinuxONE instance on the cloud.

LinuxONE servers can be configured to host the ICP software, which is a platform that integrates DevOps capabilities with cloud-optimized software. ICP on LinuxONE servers allows teams to take advantage of the IBM and open source portfolios of software by using containers and microservices in a secure cloud environment.

Deploying an ICP environment on the LinuxONE platform allows customers to take advantage of the security capabilities of IBM's unique enterprise server architecture. For example, customers can decide to encrypt all their data (in flight or at rest) by using pervasive encryption, and isolate containerized applications further by using IBM Secure Service Container for ICP (which automatically includes pervasive encryption) to create a highly secure environment.

This automatic encryption protects applications and data from attacks that attempt to gain access through privileged administrator credentials. By using these security capabilities, clients can securely build and host their own hybrid and private cloud deployments on-premises.

1.4.4 IBM Hyper Protect Services overview

Various security services are available that are hosted in the IBM Cloud.

IBM offers the IBM Hyper Protect Services built with enterprise-grade data protection, which is made possible by bringing IBM LinuxONE into IBM's global public cloud data centers.

Now, developers and clients can build, deploy, and host applications with an industry-leading data protection that encrypts information that is at rest and in flight. This technology is designed to help protect against threats inside and outside of an organization.

The IBM Cloud Hyper Protect family provides the following services as beta releases and intends to expand to include others that are crucial for providing protected cloud capabilities:

- ▶ IBM Cloud Hyper Protect Crypto-Services
- ▶ IBM Cloud Hyper Protect DBaaS

IBM Cloud Hyper Protect Crypto-Services

Keep your own keys for cloud data encryption protected in a dedicated cloud hardware security module (HSM). You also can maintain control of the key hierarchy, including the HSM master key.

Features

IBM Cloud Hyper Protect Crypto-Services include the following features:

- ▶ Control your HSM
You can keep your own keys (KYOK) by uploading your own master key that is protected by FIPS 140-2 Level 4 certified hardware.
- ▶ Customer-managed encryption
You can enable the security benefits of Bring Your Own Key (BYOK) by importing your own root of trust encryption keys, called Customer Root Keys (CRKs), into the service. With the Key Protect API, you can use a CRK to wrap (encrypt) and unwrap (decrypt) the keys that are associated with your data resources so that you control the security of your encrypted data in the cloud.

Getting started with IBM Cloud Hyper Protect Crypto-Services

IBM Cloud Hyper Protect Crypto-Services (Hyper Protect Crypto-Services) is a key management and cloud HSM. It enables you to control your cloud data encryption keys and cloud HSMs. It is the only service in the industry that is built on FIPS 140-2 Level 4-certified hardware.

Hyper Protect Crypto-Services integrates with IBM Key Protect for IBM Cloud APIs to generate and encrypt keys. The KYOK function is also enabled by Hyper Protect Crypto-Services to provide access to cryptographic hardware that is FIPS 140-2 Level 4 certified technology, the highest level attainable of security.

Hyper Protect Crypto-Services offers network addressable HSMs. For more information about security requirements for cryptographic modules, see the specification of the NIST for FIPS 140-2 Level.

For more information, see this IBM Cloud Docs [web page](#).

IBM Cloud Hyper Protect DBaaS

As business leaders look to use the cloud, enterprises in highly regulated industries often are concerned about protecting confidential and sensitive customer data. IBM is aware of public reports of unauthorized user access, encryption details, or exposure of data by internal users rising at an alarming rate. Because of the high average cost of a security breach (\$3.86 million per a 2018 Ponemon Cost of Data Breach Study, which was sponsored by IBM), it is no wonder security is at the forefront of many people.

Fully managed and highly secure databases, including PostgreSQL and MongoDB Enterprise Server, provide a high level of data confidentiality for your sensitive data in the IBM Cloud.

For more information, see the IBM Cloud [web page](#).



Introduction to security on IBM LinuxONE

This chapter provides an introduction to security on IBM LinuxONE. Also described are the specifics of IBM z/Architecture® Virtual Machine (IBM z/VM) security, and the benefits of the use of an external security manager (ESM); for example, IBM Resource Access Control Facility (RACF®) for z/VM.

z/VM virtual machines are also referred to as *guests*, *user IDs*, or *service machines*.

With IBM LinuxONE Architecture, you have many security features that you can use to secure your applications. However, you do not only set up the features; you also must customize them correctly.

Because operating systems alone cannot provide the necessary security, this chapter also provides a brief overview of hardware security features.

Note: If you must comply with the requirements of the Common Criteria Operating System Protection Profile (OSPP), you must install RACF and the Single System Image (SSI) feature because evaluation for z/VM was done only with these features enabled. For more information, see *z/VM Secure Configuration Guide*, SC24-6323.

This chapter includes the following topics:

- ▶ 2.1, “Why security matters” on page 16
- ▶ 2.2, “Hardware security features overview” on page 16
- ▶ 2.3, “Pervasive encryption” on page 17
- ▶ 2.4, “IBM LinuxONE cryptographic hardware features” on page 18
- ▶ 2.5, “Benefits of hardware crypto” on page 19
- ▶ 2.6, “Using RACF to secure your cloud infrastructure” on page 20
- ▶ 2.7, “RACF DB organization and structure” on page 22

2.1 Why security matters

Security is essential in many ways. This axiom is true for physical security, but because electronic services are more prevalent, it is evident that companies must secure and protect these services, too.

Every company that handles customer information or offers services through internet platforms must make sure that processed data is secured against all threats.

All precautions to prevent data leakage and to assure system and data integrity must be taken. It is no longer sufficient to state that data processing is secure today. You also must offer proof to auditors and comply with regulations to establish trust in your services.

Most important, you must prevent a loss of revenue and reputation because of security exposures.

Therefore, it is a preferred practice to establish the strongest security mechanisms at all levels of data processing, including the physical security of the machine rooms at your data center (controlling access to the facilities) and implementing appropriate access levels to applications, programs, data, archives, and so on. The principle of least privilege should be met at all levels.

This chapter provides guidelines about how to meet security demands in a cloud environment that is provided by z/VM and Linux on IBM LinuxONE.

2.2 Hardware security features overview

The hardware security features provide a fundamental part of the security definitions of software techniques and solutions. The available operating systems for IBM LinuxONE, z/VM, and Linux on IBM LinuxONE each use these hardware features to some degree.

Understanding IBM LinuxONE hardware and architecture are key to understanding how operating systems and applications maintain data, process, and application integrity.

Security features on the mainframe are integrated into the hardware. The following hardware security features are available:

- ▶ With the pervasive encryption, it is possible to encrypt all the data that is associated with an application or a database at once. The LinuxONE platform includes dedicated hardware, which is the one-chip encryption co-processor that is on every compute chip next to the main processor and can encrypt up to 13 GB of data per second per core.
- ▶ With the Hardware Management Console (HMC), logical partitions (LPARs) can be defined and isolated from each other. Also, all of the resources that are needed to run the operating systems are defined through LPAR profiles by the HMC. These resources are storage and processors and Direct Access Storage Device (DASD) and tape units.
- ▶ Crypto-Express Cards can encrypt session traffic and physical data on DASDs and tape. Cryptographic coprocessors are used for better performance. For more information, see 3.5.2, “z/VM Cryptographic definitions” on page 44.
- ▶ Signed microcode is applied to the hardware to ensure microcode authenticity.

z/VM provides a host of features that isolates virtual machines (VMs) (also called *guests*) from one another. This isolation is implemented in the z/VM Control Program (CP), which can be considered the kernel of the hypervisor.

Separation of guest workloads is a vital component of system integrity. It provides the foundation of the security context on which the IBM LinuxONE Integrity Statement is based. For more information about the z/VM CP, see *z/VM CP Planning and Administration*, SC24-6271.

2.3 Pervasive encryption

Data protection and security are business imperatives, and regulatory compliance is increasing in complexity. Extensive use of encryption is one of the best ways to reduce the risks and financial losses of a data breach and meet complex compliance mandates. However, implementing encryption can be a complex process for organizations. They must determine the following factors:

- ▶ What data should be encrypted?
- ▶ Where should encryption occur?
- ▶ Who is responsible for encryption?

Because the data is the new perimeter, encryption policies must cover both data in-flight and data at-rest. However, they should not require costly application changes to achieve this goal. Organizations need a transparent and consumable approach to enable extensive encryption of data in-flight and at-rest to substantially simplify and reduce the costs that are associated with protecting the data at the core of their enterprise and achieving compliance mandates.

With solutions around privileged identity management, sensitive data protection, and integrated security intelligence, IBM LinuxONE security offers the next generation of secure, trusted transactions.

Pervasive encryption is a data-centric approach to information security that entails protecting data that is entering and exiting the IBM LinuxONE platform. It involves encrypting data in-flight and at-rest to meet complex compliance mandates and reduce the risks and financial losses of a data breach. It is a paradigm shift from selective encryption (where only the data that is required to achieve compliance is encrypted) to pervasive encryption.

Pervasive encryption with IBM LinuxONE is enabled through tight platform integration that includes the following features:

- ▶ Integrated cryptographic hardware: CPACF is a co-processor on every processor unit that accelerates encryption. Crypto-Express features can be used as hardware security modules (HSMs).
- ▶ Data set and file encryption: You can protect Linux file systems by using policy-controlled encryption that is transparent to applications and databases.
- ▶ Network encryption: You can protect network data traffic by using standards-based encryption from endpoint to endpoint.
- ▶ Full disk encryption: You can use disk drive encryption that protects data at rest when disk drives are retired, sent for repair, or repurposed.
- ▶ Secure Service Container: Secure deployment of software appliances, including tamper protection during installation and run time, restricted administrator access, and encryption of data and code in-flight and at-rest.

Pervasive encryption has the following advantages:

- ▶ The ability to encrypt data by policy without application change.
- ▶ A simplified way to protect data at a much coarser scale with industry best performance.
- ▶ Greatly simplified audit, enabling clients to pass compliance audits more easily.

IBM LinuxONE excels with security features that are built into the hardware, firmware, and operating systems. The built-in features range from storage protection keys and workload isolation to granular audit capabilities, and more. The CPACF, standard on every core, supports pervasive encryption and provides hardware acceleration for encryption operations. In addition, the new Crypto-Express6S gets a performance boost on IBM LinuxONE.

Security in individual layers might be enough to keep the data integrity, confidentiality, and availability at the destination. However, it is important to secure the data while it is in transit during communication.

Some solutions can be implemented at the client side, but the organization cannot rely on client-side only security. Users might forget to update their security software, security operating system updates might unknowingly install malware on their devices that prevents the execution of the security software, or the users might not install the security software.

What the organization can do is make sure the communication between the client and the server is encrypted with a secure cryptographic protocol. New vulnerabilities are often discovered on cryptographic protocols, cipher algorithms, and protocol implementation, so the security team must be up to date about what is secure to be used, and new vulnerabilities that must be mitigated as soon as they are reported.

The encryption of data is expensive and can heavily affect performance, throughput, or CPU load of a system. IBM LinuxONE provides hardware encryption support that can be used to reduce the effect of expensive encryption operations. Because the encryption operations are offloaded to the IBM LinuxONE CPACF processor or to the Crypto-Express6S card, the performance and throughput of your workload is less affected.

2.4 IBM LinuxONE cryptographic hardware features

Servers of the IBM LinuxONE family provide two different types of hardware support for cryptographic operations: CPACF and Crypto-Express (CEX) features.

2.4.1 CP Assist for Cryptographic Function

CP Assist for Cryptographic Function (CPACF) offers a set of symmetric cryptographic functions for high-performance encryption and decryption with clear key operations for SSL/TLS, VPN, and data-storing applications that do not require FIPS 140-2 level 4 security.

CPACF is an optional feature that is integrated with the compression unit in the coprocessor in the IBM LinuxONE microprocessor core. CPACF is available on every Processor Unit that is defined as a CP or IFL.

The CPACF protected key is a function that facilitates the continued privacy of cryptographic key material while keeping the wanted high performance. CPACF ensures that key material is not visible to applications or operating systems during encryption operations. A CPACF protected key provides substantial throughput improvements for large-volume data encryption and low latency for encryption of small blocks of data.

2.4.2 Crypto-Express6S

The Crypto-Express6S represents the newest generation of the Peripheral Component Interconnect® Express (PCIe) cryptographic coprocessors, an optional feature exclusive to the IBM LinuxONE. HSMs provide the high-security cryptographic processing that is required by banking and other industries. This feature provides a secure programming and hardware environment wherein crypto-processes are performed.

Each cryptographic coprocessor includes general-purpose processors, non-volatile storage, and specialized cryptographic electronics, all contained within a tamper-sensing and tamper-responsive enclosure that destroys all keys and sensitive data on any attempt to tamper with the device. The security features of the HSM are designed to meet the requirements of FIPS 140-2, Level 4, the highest security level defined.

The Crypto-Express6S has one PCIe adapter per feature. For availability reasons, a minimum of two features is required. Up to 16 Crypto-Express6S features are supported. The Crypto-Express6S feature occupies one I/O slot in a PCIe I/O drawer.

Each adapter can be configured as a Secure IBM CCA coprocessor, a Secure IBM Enterprise PKCS #11 (EP11) coprocessor, or as an accelerator.

A cryptographic coprocessor is divided into multiple domains, also called AP queues. Each AP queue acts as an independent cryptographic device (HSM) with its own state, including its own master key. Crypto-Express6S provides domain support for up to 85 logical partitions.

Adapter management is done with the SE or the HMC by performing the following actions:

- ▶ Selection of adapter type (firmware load)
- ▶ Assignment of adapters and domains to LPARs

Note: The Trusted Key Entry (TKE) Workstation feature is required for supporting the administration of the Crypto-Express6S when configured as an Enterprise PKCS #11 coprocessor or managing the new CCA mode PCI-HSM.

For more information about pervasive encryption, see *Getting Started with Linux on Z Encryption for Data At-Rest*, SG24-8436.

2.5 Benefits of hardware crypto

The encryption of data is expensive and can heavily affect performance, throughput, or CPU load of a system. IBM LinuxONE provides hardware encryption support that can be used to reduce the effect of expensive encryption operations. Because the encryption operations are offloaded to the IBM LinuxONE CPACF processor or to the Crypto-Express6S card, the performance and throughput of your workload is less affected.

For the first time, IBM LinuxONE makes it possible for organizations to pervasively encrypt data that is associated with an entire application, cloud service, or database in flight or at rest with one click. The standard practice today is to encrypt small chunks of data at a time, and invest significant labor to select and manage individual fields.

This bulk encryption at cloud scale is made possible by a massive 7x increase in cryptographic performance over the previous IBM LinuxONE generation. This increase is driven by a 4x increase in silicon that is dedicated to cryptographic algorithms. This rate is 18x faster compared to x86 systems (that today focus on only limited slices of data) and at just five percent of the cost of comparable x86-based solutions.

A top concern for organizations is protection of encryption keys. In large organizations, hackers often target encryption keys, which are routinely exposed in memory as they are used. IBM LinuxONE can protect millions of keys (and the process of accessing, generating, and recycling them) in “tamper responding” hardware that causes keys to be invalidated at any sign of intrusion. The keys can then be restored in safety.

The IBM LinuxONE key management system is designed to meet Federal Information Processing Standards (FIPS) Level 4 standards, whereas the norm for high security in the industry is Level 2. This IBM LinuxONE capability can be extended beyond the CEC to other devices, such as storage systems and servers in the cloud. In addition, IBM Secure Service Container protects against insider threats from contractors and privileged users, provides automatic encryption of data and code in-flight and at-rest, and tamper-resistance during installation and run time.

2.6 Using RACF to secure your cloud infrastructure

If you are running applications that must meet mandatory regulations, such as the rules of the Payment Card Industry Data Security Standard (PCI DSS), you must adhere to several controls and evidences to pass auditor checks. You can meet this requirement by setting the auditing controls according to your installation’s needs, as described in 5.4, “Auditing” on page 130.

In addition to the operating system built-in security mechanisms, such as isolation of virtual storage by the z/VM CP, Resource Access Control Facility (RACF) provides ways to better control access to resources in your system. However, meeting the regulatory needs is not done by setting up only the RACF databases and defining profiles to protect resources. Your entire organization should implement a security policy and set up the RACF definitions according to a defined policy.

Implementing security processes is an ongoing process in your company and needs the full support of all managers of your organization. Implementing security processes needs much organizational work that is done with documentation processes and reviews, both of which are deeply integrated in your company’s structure. This process means a reasonable amount of work for security administrator staff and many departments of an organization.

With RACF installed, you can perform the following tasks:

- ▶ Track who uses privileged accounts; that is, MAINT and MAINT710.
- ▶ Prevent technical support user IDs and VM guests from being revoked by a password revocation policy. To do so, you define these IDs as Protected user IDs. Together with the RACF class SURROGAT **logonby** policy, you can get full information about who used the VM.
- ▶ Provide logging mechanisms (SMF records) to show the following information:
 - Who accessed what resources.
 - Which access violations occurred.
- ▶ Meet Segregation of Duty needs by separating defined Security Administrators from System Programmer staff. Principles of RACF operations

RACF provides the tools to manage user access to critical resources. RACF is an add-on software product that provides basic security for a mainframe system (examples of other security software packages include ACF2 and Top Secret, both from Computer Associates).

RACF protects resources by granting access only to authorized users of the protected resources. RACF retains information about users, resources, and access authorities in special structures that are called profiles in its database, and it refers to these profiles when deciding which users should be permitted access to protected system resources.

Modern z/VM security requires an ESM, such as the RACF for z/VM feature. This security server functions as a Policy Decision Point and Policy Enforcement Point for all security-relevant events in your virtual infrastructure (and by extension, your cloud). RACF for z/VM can be configured to handle resource authorization, privileged command access, and logon controls.

RACF provides services for authentication and authorization to resources.

To have the services of z/VM RACF available, a RACF database must be set up, and a user ID in which the RACF binary files are available must be started. In z/VM RACF, this VM is RACFVM.

Note: If RACF is installed, users' passwords are never stored in clear text in the system; instead, they are stored in encrypted form in the RACF database. For more information about encryption algorithms, see "Password encryption algorithm" on page 74.

Also, the passwords in the USER directory are no longer in effect.

With the z/VM 7.1, password encryption support for KDFAES is available. Using this algorithm provides better protection against brute-force attacks if an offline copy of the RACF database becomes exposed.

The RACF database is used to store all information about users, groups, and resources. Access to resources is controlled through entries in the following lists:

- ▶ Standard access control lists of the resource profiles
- ▶ Conditional access control lists of resource profiles (resource access is allowed only through a certain program)

Note: The preferred practice of RACF administrators is to give access rights to groups rather than users.

For more information about how to get started with RACF and how to adopt RACF definitions to your business demands for a security structure, see Chapter 4 "IBM Resource Access Control Facility Security Server for IBM z/VM, and z/VM RACF Security Server Security Administrator's Guide, SC24-6311.

2.6.1 Principle of best matching profile

RACF uses the principle of *best matching profiles* to check whether access might be granted because of the access rights that are stored in a RACF database.

A profile that covers the name of a resource is best used to check the access. The access intent must at least meet the access that is stored in the RACF profile's access list. For more information about this principle, see *z/VM RACF Security Server Security Administrator's Guide, SC24-6311*.

If you run z/VM in an SSI cluster environment, RACFVM is an identity service machine, which means it runs on every z/VM image in the cluster. To provide this service, a RACF database is needed and shared among the SSI members. The RACF database and its backup are on two distinct DASD volumes, each of which is shared in an SSI cluster. For more information about RACF databases, see 2.7, “RACF DB organization and structure” on page 22.

2.7 RACF DB organization and structure

This section describes the RACF database, how it is defined to the system, and its internal organization.

2.7.1 Database definition to the system

The RACF database is referenced by the database name table (ICHRDSNT) in the system. You can set up the RACF database by running the **RACDSF**, **RACALLOC**, and **RACINITD** RACF commands. For more information about these commands, see Chapter 4, “Operating Considerations unique to z/VM”, in *RACF Security Server System Programmer's Guide* SC24-6312.

Note: Allocation and DASD sharing options depend on the type of z/VM installation you use. Set up RACF database sharing correctly according to your system's installation, or RACF database corruption might occur. In an SSI environment, the RACF database must be shared among all members of the cluster.

More changes to the definition of RACF database devices apply if you run an IBM Geographically Dispersed Parallel Sysplex™ (IBM GDPS®) controlled system.

The number of physical extents of the RACF database is 1 by default. It is controlled through the RACF database range table (ICHRRNG), which is a load module. This table is in RACFLPA LOADLIB on the RACFVM 305 minidisk.

For more information about the RACF database range table, see Chapter 3, “RACF Customization”, in *RACF Security Server System Programmer's Guide*, SC24-6312.

2.7.2 Internal organization of RACF database specifying class options

RACF can protect the following types of resources:

- ▶ Users
- ▶ Groups
- ▶ General resources

Classes of general resources are defined in the class descriptor table (CDT). Each general resource class is defined by a unique entry in the CDT.

The CDT describes the structure of profiles for the general resource classes. If you do not comply to the settings in the CDT for the general resource class, one of the following conditions might apply:

- ▶ You cannot define the profile.
- ▶ RACF cannot determine the matching profile for the access check, which leaves resources unprotected by RACF in the system.

For example, we define a resource entry for a VMLAN VSWITCH entry by using the command that is shown in Example 2-1.

Example 2-1 RACF VMLAN definition

```
RAC RDEF VMLAN SYSTEM.VSWITCH1.010 UACC(NONE) OW(SYS1)
```

Because CDT for VMLAN defines the last qualifier as a four-digit value, RACF issues the message that is shown in Example 2-2.

Example 2-2 RACF error message

```
IKJ56702I INVALID ENTITY, SYSTEM.VSWITCH.010
```

To correct this error, ensure that you define the profile as SYSTEM.VSWITCH1.0010.

Also, the CDT is used to determine whether a RACF class can be RACLISTed or GENLISTed by running the **SETROPTS** command. RACLIST is a performance option, profiles of the classes are kept in storage, and no I/O operation occurs on the RACF database when checking on these profiles. However, changes to the profiles need an in-storage refresh of RACLISTed profiles. This process is done by running the **SETROPTS REFRESH** command.

In addition, the following CDT entry types are available:

- ▶ ICHRRCDX is the name for the IBM-supplied class entries.
- ▶ ICHRRCDE is the name for installation-defined class entries.

Note: Do not delete or modify any of the class entries in the IBM-supplied load module ICHRRCDX.

For more information about IBM-supplied class entries, see Appendix B, “Description of the RACF classes”, in *RACF Security Server System Programmer’s Guide*, SC24-6312.



IBM z/VM hypervisor

This chapter describes the security aspects of z/VM facilities. It also introduces how the z/VM hypervisor can provide security in its virtualization environment on IBM LinuxONE and how it can be improved with the installation of an external security manager (ESM), such as IBM Resource Access Control Facility (RACF).

Protecting information from unintended use is one key element of a secure IT environment. The following methods can be used to ensure privacy of information:

- ▶ Access control
- ▶ Encryption methods

Access control mechanisms determine who has the right to access particular information or data. The access control mechanisms then verify who accesses the information (*authentication*) and whether they have the right to access this information (*authorization*).

Cases exist in which proper access control cannot be ensured in all situations, especially if data is stored on movable media and also when data is transferred through a network that might not be protected. It is not possible to ensure that no unintended access to data occurs while it is stored or transferred through a network. The only way to protect such information is by using encryption methods.

This chapter includes the following topics:

- ▶ 3.1 “Virtualization” on page 26
- ▶ 3.2 “z/VM hypervisor and LinuxONE servers” on page 27
- ▶ 3.3 “Device management” on page 38
- ▶ 3.4 “Securing the data” on page 38
- ▶ 3.5 “Securing your communication” on page 42
- ▶ 3.6 “z/VM connectivity” on page 50
- ▶ 3.7 “Remote Spooling Communications Subsystem” on page 52

3.1 Virtualization

Organizations today are challenged to deliver improved information services to the business within severely constrained budgets. Many organizations are considering cloud computing models to gain benefits, including reduced cost and a shift in cost profile from capital to operating expense. Other benefits, such as scalability, improved flexibility and agility, and more efficient utilization of human and technology resources are also attractive.

Virtualization is the key for cloud as it creates the infrastructure of resources that are used by Cloud Computing. By using a virtual infrastructure, you can create a cloud by pooling virtual resources, orchestrating them by using management and automation software, and creating a self-service portal for users.

Virtualization is technology that allows you to create multiple simulated environments or dedicated resources from a single, physical hardware system. An operating system (called a *hypervisor*) connects directly to that hardware and allows you to split one system into separate, distinct, and secure environments that are known as virtual machines (VMs). These VMs rely on the hypervisor's ability to separate the machine's resources from the hardware and distribute them.

From the perspective of the enterprise, virtualization developed into a significant strategy for organizations that are facing rising costs, and do not want to affect service levels. The increasing need for agility in market response is also pushing more to implement virtualization, with more production VM images being deployed every day and development and test images deployed in minutes.

Virtualization provides a secure environment with isolation and sharing resources that allows a single platform to work as though multiple hardware environments were used. Because of the global economy, customers are driven by the business marketplace, and that pushes organizations to continue searching for higher efficiencies and better usage of IT resources.

Key growing cloud security concerns include the following examples:

- ▶ Securing highly virtualized environments from targeted threats and attacks
- ▶ Enabling secure user collaboration and protecting the data (isolation and sharing)
- ▶ Provisioning and de-provisioning (users and technical components) rapidly
- ▶ Losing direct control of security compliance and privacy parameters

3.1.1 Virtualization benefits

Virtualization provides the following benefits:

- ▶ Consolidation to reduce hardware costs

Virtualization enables clients to access and manage resources efficiently to reduce operations and systems management costs while maintaining needed capacity. It also enables clients to have a single-server function as multiple virtual servers.

- ▶ Optimization of workloads

Virtualization enables clients to respond dynamically to the application needs of its users. Virtualization also can increase the use of resources by enabling dynamic sharing of resource pools.

- ▶ IT flexibility and responsiveness

Virtualization enables clients to have a single, consolidated view of, and easy access to, all available resources in the network, regardless of location. It also enables clients to reduce the management of their environment by providing emulation for compatibility, improved interoperability, and transparent change windows.

By adding any of these virtualization technologies to your environment, you create an on-demand, secure, and flexible infrastructure that is prepared to handle workload changes in your environment. IBM virtualization is the answer for the Cloud because it perfectly adheres to the hardware server and uses the LinuxONE functionalities with efficiency, high throughput, and security that is inherent of this platform.

3.1.2 Hardware virtualization

The following facilities are available in LinuxONE servers and are used to divide the hardware into LPARs. They are classified as type 1 hypervisors, are run directly on the hardware, and are known as “native” or “bare metal”:

- ▶ IBM PR/SM™ is the facility on LinuxONE servers that provides another layer of virtualization. It is a hypervisor that is classified as type 1 and allows multiple LPARs to share physical resources, such as processors, memory, channel paths, and DASDs.
- ▶ Dynamic Partition Manager (DPM) is a new administrative mode that was introduced to LinuxONE servers. A system can be configured in DPM mode or PR/SM mode. The mode is enabled before system power-on reset (POR).

DPM provides the following advantages:

- Fast
Much faster than managing with HCD and HCM (from hours to minutes).
- Easy
Intuitive user interface. No need for multiple administrators with different skills or tools.
- Powerful
The same efficient PR/SM hardware virtualization, without the complexity. It supports dynamic configuration changes with just a few mouse clicks. It provides a foundation for “bare metal” cloud.

3.2 z/VM hypervisor and LinuxONE servers

z/VM is the mature virtualization solution of IBM. It is reliable and flexible, and shares the resources to support thousand of guest servers with a high performance. As a hypervisor operating system, its security does not differ from the security of any other operating system on a server. However, the virtual infrastructure relies on the security of the hypervisor, so protecting the z/VM hypervisor often prevents attempts to breach the security of the operating system and compromises to the integrity of the operating system and data.

The merger of z/VM and Linux on LinuxONE is perfect because z/VM can have hundreds of Linux servers that are running harmoniously while z/VM manages all of the resources of the LPAR with the benefit of improved price performance as workloads grow.

A fundamental strength of z/VM is the ability for VMs to share system resources with high levels of resource usage. z/VM V7.1 provides even greater levels of extreme scalability, security, and efficiency to create opportunities for cost savings, while providing a robust foundation for cognitive computing on the LinuxONE platform.

When multiple Linux servers run on LinuxONE, each Linux system includes dedicated access to a defined portion of the LinuxONE machine as provided by the hypervisor by using a technique that is known as *timesharing*. Each Linux instance runs in its own VM whose characteristics (for example, memory size and number of CPUs) define the hardware that Linux sees. The allocation and tuning controls in z/VM specify how real hardware resources are allocated to the VM.

Although each guest (a Linux server) can have its own security configuration and faces its own threats, it is essential to protect the hypervisor as an equally important part of an overall end-to-end security policy because important activities, such as creating, changing, and removing VMs are performed at the hypervisor level. Protecting the guests and not the hypervisor is similar to locking all of the windows of your home and then leaving the front door open.

Performing z/VM maintenance is part of the system administrator role. It is important to apply service to your z/VM system to ensure that the latest security measures are in place; therefore, installing the corrections when they are released decreases the time frame that the vulnerability can be used.

In addition to operating system setup and customization for security, monitoring the hypervisor for signs of compromise helps you promptly respond to a threat. Use monitoring tools to help monitor the hypervisor and review the hypervisor logs for suspicious activities, both of which make the work of the hypervisor system administrator easier.

In addition to operating system setup and customization for security, monitoring the hypervisor for signs of compromise helps you promptly respond to a threat. Use monitoring tools to help monitor the hypervisor and review the hypervisor logs for suspicious activities, both of which make the work of the hypervisor system administrator easier.

Note: z/VM supports more VMs in a single footprint with more excellent service levels than any other solution. It also allows a user to scale up the system capacity without requiring more support personnel.

3.2.1 z/VM 7.1 overview

z/VM V7.1 supports IBM LinuxONE servers and Red Hat, SUSE, and Ubuntu Linux distributions. Support for simultaneous multithreading (SMT) technology extends per-processor, core capacity growth beyond single-thread performance for Linux guests that are running on an IBM Integrated Facility for Linux (IFL) specialty engine on IBM LinuxONE servers.

Its multithreading technology support provides more price and performance benefits over previous hardware generations. It also can meet workload requirements transparently. Improvements that are made in the areas of reliability, availability, and serviceability allow low-end devices, such as IBM Storwize® V7000, V840, and V9000, to be attached to a z/VM host, which removes the need for an IBM SAN Volume Controller.

z/VM V7.1 is a supported environment that uses IBM Dynamic Partition Manager for Linux-only systems. This configuration simplifies system administration tasks for a more positive experience. Also, the use of IBM Wave with z/VM can greatly simplify the task of administering a z/VM environment.

By using z/VM as a hypervisor for Linux workloads, you can extend the business value of IBM LinuxONE technology across the enterprise by integrating applications and data, while providing exceptional levels of availability, scalability, security, and operational ease.

Integration of z/VM SSI for continuous operation

z/VM single system image (SSI) is included in the base of z/VM V7.1 at no extra cost. Integrating and making SSI available at no charge is intended to help more clients reduce or shorten planned outages of their Linux workloads as they adopt the z/VM Continuous Delivery model for their z/VM systems. SSI includes live guest relocation and single system maintenance to give clients a mechanism to host Linux virtual server images without suffering interruptions as they apply updates to their z/VM systems.

Note: For more information about z/VM 7.1 and its functionalities, see [this website](#).

3.2.2 Single System Image overview

The architecture of Linux solutions on IBM LinuxONE changed dramatically when z/VM Single System Image (SSI) is used with live guest relocation (LGR).

An SSI cluster is a multi-system environment on which the z/VM systems can be managed as a single resource pool and guests can be moved from one system to another while they are running. Each SSI member is a z/VM logical partition (LPAR) connected through channel to channel (CTC) connections, and the z/VM SSI cluster consists of up to four z/VM systems in an Inter-System Facility for Communications (ISFC) collection. CTC connections are physical connections and because the channels are isolated from the outside world, the traffic does not need to be encrypted.

Each z/VM system is a member of the SSI cluster and is self-managed by the z/VM Control Program (CP). All members can access shared DASD volumes, and the same Ethernet LAN segments and storage area networks (SANs).

Note: Starting with z/VM 7.1, SSI is included in the base hypervisor at no extra cost. Integrating and making SSI available at no charge helps more clients reduce or shorten planned outages of their Linux workloads as they adopt the Live Guest Relocation (LGR) for their z/VM systems.

Live guest relocation

With the IBM z/VM SSI, a running Linux on IBM LinuxONE can be relocated from one member z/VM system to any other, a process known as LGR. LGR occurs without disruption to the business and provides application continuity across planned z/VM and hardware outages and flexible workload balancing that allows work to be moved to available system resources.

A running virtual server might need to be relocated for one of the following reasons:

- ▶ Increased flexibility for planned outages
- ▶ Maintenance of hardware or software
- ▶ Fixing performance problems
- ▶ Management and balancing of workloads

Relocating virtual servers can be useful for load balancing and for moving workload off a physical server or member system that requires maintenance. After maintenance is applied to a member, guests can be relocated back to that member, which allows you to maintain z/VM and keep your Linux on IBM LinuxONE virtual servers available.

Note: Currently, Linux server running in z/VM is the only guest environment that is supported for relocation.

For more information about SSI and LGR, see the following resources:

- ▶ *z/VM CP Planning and Administration version 7 release 1*, SC24-6271
- ▶ *An Introduction to z/VM Single System Image (SSI) and Live Guest Relocation (LGR)*, SG24-8006
- ▶ *Using z/VM 6.2 Single System Image (SSI) and Live Guest Relocation (LGR)*, SG24-8039
- ▶ Chapter 7 of *The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3*, SG24-8147.

Changes for SSI in the USER directory

The z/VM user directory (or user registry) describes the configuration and operating characteristics of each virtual machine that can be created by CP. A z/VM user directory exists in two forms: a source form that consists of one or more CMS files, and an object form (which is created from the source) on a CP-formatted disk.

The source form of the user directory consists of directory statements that define the CP volume on which the object directory is created and the characteristics of each virtual machine that runs on z/VM system.

This section provides an overview of the following definitions in the z/VM directory for guests with single configuration and multiple configurations (see Figure 3-1 on page 31):

▶ Single-configuration VM

A single-configuration VM definition consists of a user entry and any included profile entry. For example, you can specify a single-configuration virtual machine as EDI and log on to a z/VM system as EDI. In an SSI cluster, the VM can be logged on to only one SSI member at a time. Your Linux guests are always defined as single users.

▶ Multi-configuration VM

A multi-configuration VM definition consists of an identity entry and all associated subconfiguration entries (**SUBCONFIG** in **BUILD ON** z/VM Directory Manager (IBM DirMaint™) statement). In an SSI environment, this VM definition allows multiple instances, which enables the user ID to be logged on concurrently to multiple members of the SSI cluster. Each of these VM instances can have a different configuration as minidisks in each LPAR member, and so on.

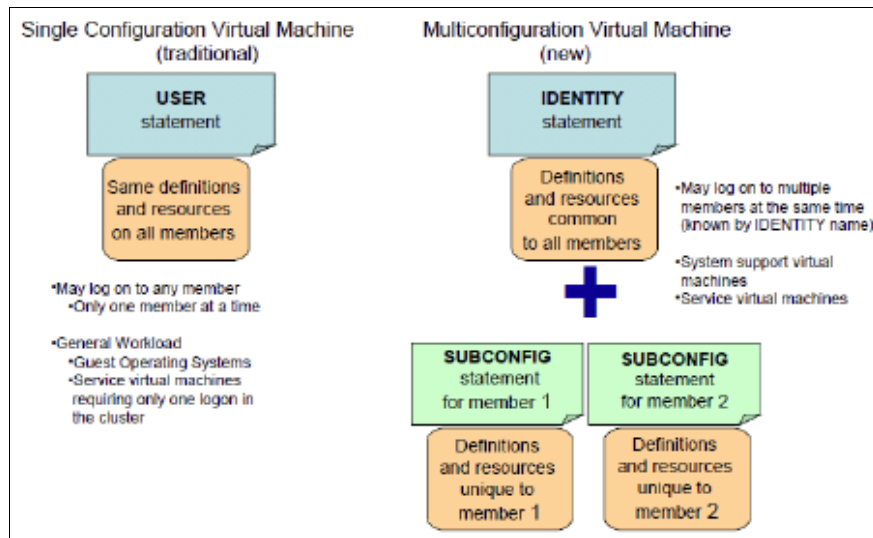


Figure 3-1 User definitions as a User or Identity

USERS are relocatable and can access the same resources no matter to which par they go. An IDENTITY is restricted to a single cluster member and might access private resources of each z/VM member.

Note: A z/VM SSI cluster uses a single source directory to define VMs on the system. However, separate object directories are built on each member node of the cluster. As a result, care must be exercised when changing VMs on the system so that a new object directory is compiled on each member of the cluster at the same time.

3.2.3 Security settings in an SSI cluster

The following preferred practices can be used to make your SSI cluster more secure and compliant with security rules:

- ▶ A Single Configuration VM can log on to only one member of the cluster.
- ▶ A Multi-Configuration Virtual Machine can have a different definition on each system.
- ▶ A user ID's privilege classes and passwords are the same on every system.

A common source directory definition is available.

- ▶ The cluster maintains a *single security context* for the entire system.

An ESM extends these capabilities, even in stand-alone systems.

- ▶ Consider the use of Relocation Domains.

Relocation Domains can be built so that guests can be constrained to certain cluster members. It is a way of building security zones into the SSI by constraining data flow (where the data is the server).

A z/VM system is secured by using the security features of the LinuxONE hardware by maintaining compliance to security policy within operating practices. The system administrator must lead the way in following security standards and guidelines.

When the RACF feature for z/VM is installed, it can be configured to control functions normally being checked in the directory for authorization. RACF can control the password field, minidisk access, spool files, and commands privileges.

A preferred practice to extend the z/VM environment is installing an external security manager (ESM), such as IBM RACF for z/VM or other ESM product to maximize your security.

3.2.4 Controlling the System Operator

The System Operator is a highly privileged VM that runs under z/VM. It includes all CP defined privilege classes (and access to every command), and it has special authorities over the hypervisor. It also receives informational and error messages from the components of z/VM as they occur. This user ID (most commonly OPERATOR) should be given special protections, which are described in this section.

Controlled area

Run the system operator in a physically controlled environment; for example, in a machine room or in an operator's work area. Provide proper access control through badges for authorized personnel entry only.

Automation

Set up an automation environment so that the operator close console files daily so that operator logs are ready for archiving processes. By using system user IDs, set the observer as TCP/IP, IBM DirMaint on the operator user ID.

Log on by definition

z/VM can define logon to a system by privileged user IDs, such as Operator, Maint, and Maint710.

RACF definition

With RACF, define the operator user ID as protected and enable surrogate logon processing by defining the appropriate RACF profile. Give access to surrogate profiles only to operating staff and perhaps system programmers.

Note: With RACF installed, set up an observer for operator user ID (by using Performance Toolkit for z/VM) to get an option of scrolling through the events that occurred in the past. If you do not set up this configuration, all RACF messages like ICH408I are directed to operator.

Because the operator's console is spooled, you must close the operator console file and scroll through to see the history of system events to check the most recent messages. Consider the use of a tool that helps you manage the console log of the operator daily. The archiving of console logs can then be done by z/VM when you sent the system consoles to a repository user ID or by transferring it to other systems.

3.2.5 System Configuration file

The System Configuration file is one of the most important files of z/VM. It contains operating characteristics, such as the layout of the system residence disk, real storage, and I/O devices configuration and the description of other resources that are available to the system.

The system configuration file is on a partition of a volume that is called *minidisk* and it is allocated as PARM. This minidisk is under user ID PMAINT, and its address is CF0. The file is called SYSTEM CONFIG by default, although you can change the name in your installation.

The file is read at IPL time by the CP program that uses the statements that are contained in the file to configure the system.

Note: As a best practice, always run a CPSYNTAX check after modifying SYSTEM CONFIG to check whether errors exist on this file.

The statements that are contained in the configuration file that are relevant to security are summarized next.

DEFINE COMMAND

Use the **DEFINE COMMAND** or **CMD** statement to define a new CP command or a new version of a CP command on the system during initialization.

DEFINE LAN

Use the **DEFINE LAN** statement to create a guest LAN that can be shared among virtual machines on the same VM system. Each guest LAN segment is identified by a unique combination of owner ID and LAN name. A VM user can create a simulated network interface card (NIC) and connect it to this LAN segment.

DEFINE VSWITCH

Use the **DEFINE VSWITCH** statement to create a CP system-owned switch (a virtual switch) to which VMs can connect. Each switch is identified by a *switchname*. A z/VM user can create a simulated QDIO NIC and connect it to this switch with the **NICDEF** directory statement. Under the **DEFINE VSWITCH** statement, the **VLAN** parameter is important if you want to isolate guests subnets based on VLAN IDs.

DISABLE COMMAND

Use the **DISABLE COMMAND** or **CMD** statement to prevent CP from processing requests for the specified CP command during and after initialization.

ENABLE/DISABLE DIAGNOSE

Use the **ENABLE/DISABLE DIAGNOSE** statement to permit/prevent CP from processing requests for one or more locally developed DIAGNOSE codes during and after initialization.

ENFORCE_BY_VOLId

Use the **ENFORCE_BY_VOLId** configuration statement to enforce attachment of DASD devices by their VOLIDs on the **ATTACH** command.

FEATURES

Use the **FEATURES** statement to set certain attributes of the system at system initialization.

JOURNALING

Use the **JOURNALING** statement to tell CP whether to include the journaling facility, whether to enable the system being initialized to set and query the journaling facility, and what to do if someone tries to log on to the system or link to a disk without a valid password.

Note: Journaling is not a sufficient replacement for ESM auditing, which is done by RACF.

MODIFY COMMAND

Use the **MODIFY COMMAND** or **CMD** statement to redefine an existing CP command on the system during initialization.

MODIFY LAN

Use the **MODIFY LAN** statement to modify the attributes of a guest LAN during initialization.

MODIFY VSWITCH

Use the **MODIFY VSWITCH** statement to modify the attributes of a virtual switch.

PRIV_CLASSES

Use the **PRIV_CLASSES** statement to change the privilege classes authorizing the following CP functions:

- ▶ Logging on as the primary system operator
- ▶ Intensive error recording
- ▶ Using the read function of the CP IOCP utility
- ▶ Using the write function of the CP IOCP utility
- ▶ Specifying the default user class

MODIFY PRIV_CLASSES

Use **MODIFY PRIV_CLASSES** to change the privilege classes that are authorizing the following CP functions:

- ▶ Logging on as the primary system operator
- ▶ Intensive error recording
- ▶ Using the read function of the CP IOCP utility
- ▶ Using the write function of the CP IOCP utility
- ▶ Specifying the default user class.

SYSTEM_USERIDS

Use the **SYSTEM_USERIDS** statement to specify user IDs that perform special functions during and after IPL. These functions include accumulating accounting records, system dump files, EREP records, and symptom records, and specifying the primary system operator's user ID and disconnect status.

USER_DEFAULTS

Use the **USER_DEFAULTS** statement to define default attributes and permissions for all users on the system.

Password suppression

Password suppression prevents any password from being visible on a user's screen. Suppression of visibility of passwords is mandatory to be compliant with your corporate security policies. To enable password suppression, place the following statement in the **SYSTEM CONFIG** file:

```
FEATURES PASSWORDS_ON_CMDS AUTOLOG NO LINK NO LOGON NO
```

Preventing users of T-disks and minidisks from seeing residual data

You must ensure that each time the system assigns T-disk space, it clears the space of all residual data. To ensure that this process occurs, place the following statement in the **SYSTEM CONFIG** file:

```
FEATURES ENABLE CLEAR_TDISK
```

Before the minidisk is released, it must be formatted to clear it of any residual data.

Note: For more information about the syntax and usage for the system configuration file, see *z/VM CP Planning and Administration*, SC24-6271.

3.2.6 Addressing password security

The passwords in a standard z/VM system are default passwords that are defined by the installation process. Before moving your system into production, change those passwords immediately to be compliant with your corporate security policies.

It might be mandatory based on your company policy, industry, or government regulations to change the following two types of passwords in the USER DIRECT file:

- ▶ **userid:** The password that is required to log on.
- ▶ **minidisk:** The minidisk password, which gives access to read, write, and multiple.

Changing that password can be done manually by using XEDIT, which is the z/VM text editor, or by using a macro to automate the process. Alternatively, a directory management product, such as IBM DirMaint, can be used.

Because manually changing the password is a tedious and error-prone process, make a backup copy of the USER DIRECT file only after the default passwords are changed.

Special passwords

The following special passwords in the User Direct file have specific functions:

NOLLOG	When the user ID is set with NOLLOG, it cannot be used to log in to a z/VM system until you set another password. As a preferred practice, set all unused VMs to NOLLOG.
AUTOONLY	The user ID starts running only when you issue the xautolog or autolog commands. You cannot issue logon or logoff for this user ID.
LBYONLY	This user ID can be logged on only by issuing the logon by command. You cannot log on this user ID with the logon command.

RACF control of passwords supersedes any password definitions in the CP User Directory, except the special passwords listed here. For more information, see “Password and password phrases rules” on page 115.

3.2.7 Implementing CP LOGONBY

The **CP LOGONBY** directory statement designates up to eight VMs to have access to another VM user ID. This function was originally a DirMaint implementation and was added to VM several releases ago. The **CP LOGON BY** command allows authorized VMs to log on to a shared VM by using their own password. This command is useful when several VM system supports must share the MAINT user ID, but only one person can be logged on at a time.

To fully understand this command, you must become familiar with the following terms:

- ▶ **Shared user:** A user ID that can be logged on to by a different user.
- ▶ **Surrogate user:** A person logging on to the shared user ID.
- ▶ **Direct logon:** A traditional logon, in which you log on to your own user ID.
- ▶ **Shared logon:** A logon in which a surrogate user uses the **BY** option of the **LOGON** command to log on to a different user ID.

The implementation of **CP LOGONBY** can be done updating the user directory of the user that is intended to be used as the shared user with the **LOGONBY** entry. In Example 3-1 on page 36, user MAINT is defined to be shared and user EDI is defined as one of its surrogate users.

Example 3-1 User directory of a shared user ID

```
USER OP1 XXXXXXXX 64M 96M BG
  INCLUDE IBMDFLT
  IPL CMS PARM AUTOOCR
  LOGONBY EDI KLAUS KAREN MAC
```

Now, the user EDI can use their password to log on to the OP1 shared ID, as shown in Example 3-2.

Example 3-2 Logon using LOGONBY

```
L OP1 BY EDI
ENTER PASSWORD (IT WILL NOT APPEAR WHEN TYPED):
z/VM Version 7 Release 1.0, Service Level 1801 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES: 0001 RDR, NO PRT, NO PUN
LOGON AT 09:46:46 EDT WEDNESDAY 06/12/19
z/VM CMS 29 002 01/21/19
```

It is possible to define up to eight users as surrogates of a shared user by using **CP LOGONBY**. This task can be done adding the users in the same **LOGONBY** statement of the shared user ID. Example 3-2 is an example of user MAINT being defined as surrogate of OP1.

The way the directory is defined in our examples makes it possible for user ID OP1 to be logged on by using its directory password. This configuration affects the accountability of a shared user ID because any person that knows the shared user ID password can log on directly to it.

To avoid this situation, use the keyword **LBYONLY** on the shared user ID password. It is not possible to log on the shared user ID by using the directory password. In fact, if a logon on the shared user ID is attempted, CP returns an error message and permits only to log on by that user ID, as shown in Example 3-3.

Example 3-3 Using LBYONLY statement to avoid direct logon to the shared ID

Shared user directory with the LBYONLY statement:

```
USER OP1 LBYONLY 64M 96M G
  INCLUDE IBMDFLT
  IPL CMS PARM AUTOOCR
  LOGONBY EDI WILLIANR
```

a) Tentative directly log on to OP1:

```
L OP1
HCPLGA050E LOGON unsuccessful--incorrect userid and/or password
```

b) Log on op1 by using the surrogate user ID:

```
L OP1 BY EDI
ENTER PASSWORD (IT WILL NOT APPEAR WHEN TYPED):
z/VM Version 7 Release 1.0, Service Level 1801 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES: 0001 RDR, NO PRT, NO PUN
LOGON AT 09:46:46 EDT WEDNESDAY 06/12/19
z/VM CMS 29 002 01/21/19
```

This command can be extended by using the SURROGAT class in RACF for z/VM. For more information, see 4.3 “RACF management processes” on page 88.

3.2.8 Role-based access controls and CP privilege classes

In the z/VM system of privilege, a user either can have no privileges or can be assigned to one or more *privilege classes*. Each privilege class represents a subset of Control Program commands that the system permits the user to enter. Each privilege class, sometimes called *CP privilege class*, is defined around a particular job or set of tasks, which creates an area outside of which the user cannot go. It is common for a user to be assigned to more than one CP privilege class. Users cannot enter commands in privilege classes to which they are not assigned.

Note: Any user, except those users with NO PRIVILEGE or CP privilege class G, is considered part of the configuration but is not necessarily considered trusted.

It is also possible to create privilege classes that meet the enterprise security policy according to the roles that are described in it, as described in .

The following CP privilege classes are available:

Privilege class A	The primary system operator. The system operator is among the most powerful and privileged of all z/VM users. The system operator is responsible for the system’s availability and its resources. The system operator also controls accounting, broadcasts messages, and sets performance parameters.
Privilege class B	The system resource operator. The system resource operator controls the allocation and de-allocation of real resources, such as memory, printers, and DASD. The system resource operator does not control any resource that is already controlled by the system operator or the spooling operator.
Privilege class C	System programmer. The system programmer updates the functions of the z/VM system and can change real storage in the real machine.
Privilege class D	Spooling operator. The spooling operator controls spool files and real unit record devices, such as punches, readers, and printers.
Privilege class E	System analyst. The system analyst has access to real storage and examines dumps to make sure that the system is performing as efficiently and correctly as possible.
Privilege class F	IBM service representative. The representative of IBM who diagnoses and solves problems by examining and accessing real input and output devices and the data they handle.
Privilege class G	CMS general user. This is the most prevalent and innocuous of the CP privilege classes. The commands that privilege class G users can enter affect only their own VMs.

Privilege classes A, B, C, D, E, and F should be granted to only human users and VM workloads after careful consideration regarding the scope of responsibility. For example, users with privilege class B or C can modify an installation's system of CP privilege. Users with privilege class C can enter the **cp store host** command that alters real storage. Privilege class G users have the power to modify only their own VMs; they have little security relevance and cannot violate the security policies of the system.

In the CP, each level of privilege is discrete and not predicated on others. Furthermore, each privilege class has a subset of commands and they are related to one or more function types (subsets of users).

3.3 Device management

The following methods can be used to define I/O devices to the CP during IPL:

- ▶ Enable the CP dynamically sense devices.
- ▶ Use **RDEV** statements or **EDEV** statements in the system configuration file.
- ▶ Use **RDEVICE** macroinstructions in the HCPRIO ASSEMBLE file.
- ▶ Use the Hardware Configuration Manager (HCM) and Hardware Configuration Definition (HCD) or Dynamic Partition Manager (DPM) to define the devices.

Typically, CP senses the devices. Only devices that require more definition have an **RDEVICE** or an **EDEVICE** statement in the system configuration file.

Note: For more information about defining real devices, see Chapter 5, “Defining I/O devices”, in *CP Planning and Administration*, SC24-6271.

The capabilities support of real devices is done by CP on behalf of the virtual guests, which means to the virtual guests the device is transparent in use without having access to the physical device. CP provides the system services for the device, including error recovery for guest DIAGNOSE I/O requests and a full command set for the device. Devices can be dedicated to just one guest or shared among multiple guests (which is done for DASD).

If a device supports dedicated-only use by a single guest, this device must be logically attached to a single guest at any one time. The guest must be fully capable of running with the device. CP does not supply DIAGNOSE I/O services to the guest.

3.4 Securing the data

The protection and securing of an organization's information is considered part of the foundation for business success. Ensuring strong security protection of your data is mandatory and should be deployed with a careful plan and overall understanding about the security and business requirements that your organization needs.

Note: As a starting point for defining your security policies, a smart decision is to begin with a closed security police and grant access and privileges as the business requires.

3.4.1 Securing your minidisks

The z/VM system is designed to permit access to the minidisks when you provide the correct link password that is defined on the z/VM directory. Another method is to include the link in your user direct definition. Because this environment is a controlled environment, it sounds like a secure approach, but only with an appropriate external security manager (ESM) to make your configuration resilient and less prone to error.

Note: As a best practice, change all the default passwords in the USER DIRECT file to avoid unauthorized access on the production system.

3.4.2 Encrypting z/VM page volumes

Encrypted Paging improves z/VM® system security by using IBM LinuxONE hardware to encrypt guest page data. Ciphering occurs as data moves from active memory onto a paging volume that is owned by CP, such as ECKD and SCSI devices. This configuration makes customer data defensible from an attack and from a breach of volumes, even in cases where a system administrator has unintended access to those volumes.

Encryption is limited to guest pages and virtual-disk-in-storage (VDISK) pages that are written by the CP paging subsystem to paging extents. This includes pages on an NSS/DCSS that was loaded.

The following types of pages (also written by the CP paging subsystem) are not encrypted:

- ▶ Spool files
- ▶ Directory pages
- ▶ Minidisk data to a mapped minidisk pool (established via the MAPMDISK interface)
- ▶ Minidisk cache pages
- ▶ CP page tables (PGMBKs)

Note: Encrypted Paging requires that the TRNG facility of CPACF of the IBM LinuxONE is enabled for the system.

Use the CP QUERY CRYPTO command to verify whether CPACF is enabled by issuing the command as shown in Example 3-4. It shows the status of the cryptographic hardware and the AP (AdjunctProcessor) of the Crypto-Express adapter.

Example 3-4 Query crypto to check whether it is enabled

```
Q CRYPTO
Crypto Adjunct Processor Instructions are installed
Ready; T=0.01/0.01 12:16:29
```

Note: Encrypted Paging is available starting with LinuxONE RockHopperII hardware and it is not supported on earlier machines.

Enabling z/VM encrypted paging

The following CP commands are available to query or set the encryption level:

- ▶ SET ENCRYPT

Sets dynamically enabled z/VM encrypted paging, as shown in Example 3-5 on page 40.

Example 3-5 Setting encrypt paging on or off

set encrypt paging on

Encrypt Paging set on to algorithm AES256

Encrypt Paging Settings:

Currently: On AES256

At IPL: Off

Ready; T=0.01/0.01 12:56:22

set encrypt paging off

Encrypt Paging set off

Encrypt Paging Settings:

Currently: Off

At IPL: Off

Ready; T=0.01/0.01 13:00:39

► **QUERY ENCRYPT**

z/VM encrypted paging status can be queried, as shown in Example 3-6.

Example 3-6 Querying z/VM encrypted paging status

query encrypt

Encrypt Paging Settings:

Currently: Off

At IPL: Off

Ready; T=0.01/0.01 13:01:56

Note: The encryption can be activated at IPL time by including the ENCRYPT statement in the SYSTEM CONFIG file.

Using encrypted paging

Consider the following points regarding the use of encrypted paging:

- Make sure that CPACF is enabled on your LinuxONE server.
Support requires CPACF (no-charge Feature 3863) to be enabled on z14 hardware or later.
- Set ENCRYPT PAGING ON in System Configuration or use CP SET ENCRYPT PAGING.
- Protected the ephemeral key (of selected algorithm) that is generated by CP for the lifetime of the system for all guests. No key rotation mechanism exists in this PTF.
- Support is available in OFF (default), ON, and REQUIRED modes.
Per sponsor feedback on priorities, changing the algorithm in first deliverable requires an IPL.
- To prevent against timing attacks, TRSOURCE is not be permitted in the keygen section of the IPL processes.
- One key per z/VM partition; no SSI dependencies.
Performance considerations for guest relocation include reenciphering paging data.
- A mandate for 100% encryption uses ENCRYPT PAGING ON (at minimum) at IPL:
 - ENCRYPT PAGING ON provides the function, but can be dynamically toggled
 - ENCRYPT PAGING REQUIRED includes some usability concerns
 - Dynamic support can enable compliance, but proving it is difficult (draining volumes)

SSI and LGR implications with encrypted paging

Encrypted paging does not need to be enabled on all members of a Single System Image cluster.

Ephemeral keys are not shared; one ephemeral key exists per member. When a guest is relocated, its pages are decrypted before they are relocated to the target member. The target member reencrypts the guest's pages by using its own ephemeral key.

Relocation domains can be defined per guests security requirements, such as the following examples:

- ▶ Access to hardware facilities, such as LinuxONE CPACF
- ▶ Encrypted paging in the hypervisor (requires LinuxONE partitions)

Important encrypted paging recommendations

Consider the following points:

- ▶ Test Encrypted Paging with ON before switching to REQUIRED.
- ▶ Consider one of the following options:
 - Switching from ON to REQUIRED in AUTOLOG1 (during System IPL)
 - SET ENCRYPT PAGING REQUIRED on a COMMAND statement for OPERATOR
- ▶ Have a backup System Configuration file (with setting ON) for emergency purposes.
- ▶ Double-check DR plans for the hardware feature availability of z/VM systems.

Note: If you configured REQUIRED on a system that does not support the feature, your system does not IPL.

3.4.3 Securing GUEST LANS and virtual switches

z/VM Virtual Switch supports access ports as USER-based or PORT-based. It can be VLAN-unaware, and the VSWITCH handles all VLAN tagging and trunk ports when it is VLAN-aware and processes its own VLAN tagging.

Note: The default configuration of the XCATVSW2 switch that is used by CMA defines it as VLAN-unaware.

Access to VLANs is controlled by the **GRANT** option of the CP **SET VSWITCH** command (**MODIFY VSWITCH** in **SYSTEM CONFIG**). For a user, a set of VLANs can be granted on a VSWITCH by listing them in the **VLAN** parameter. If more than one VLAN is specified, the **PORTTYPE** parameter must also be set to **TRUNK**. If a list of VLANs is given but **PORTTYPE ACCESS** is used, an error occurs, as shown in Example 3-7.

Example 3-7 SET VSWITCH GRANT with multiple VLANs and PORTTYPE ACCESS

```
set vswitch vlantst grant tcpip vlan 10 20 30
HCPSWS2847E PORTTYPE ACCESS is not allowed when the user is authorized
HCPSWS2847E for more than one VLAN
```

Note: Guest LANs are discouraged because they are more cumbersome to configure and less secure than a virtual switch.

3.5 Securing your communication

Security in individual layers might be enough to keep the data integrity, confidentiality, and availability at the destination, but it is important to secure the data while it is in transit during communication.

Some solutions can be implemented at the client side, but the organization cannot rely on client-side only security. Users can forget to update their security software or security operating system updates can unconsciously install malware on their devices that prevents the execution of the security software, or the users do not install the security software.

What the organization can do is make sure the communication between the client and the server is encrypted with a secure cryptographic protocol. New vulnerabilities are often discovered on cryptographic protocols, cipher algorithms, and protocol implementation, so the security team must be up to date about what is secure to be used, and new vulnerabilities that must be mitigated as soon as they are reported.

The IT infrastructure inside the organization is the responsibility of the organization, so all means to avoid security breaches are valid to protect the information. A well-planned network infrastructure also helps secure the data communication. The first point of contact with the internet should be the network security system. It controls the incoming and outgoing traffic to the organization's network based on the application set of rules.

Separating the network into layers helps protect the information. Therefore, during network infrastructure planning, consider at least a layer for a DMZ, a layer for the web servers, a layer for the application servers, and a layer for database servers. This is not a rule and can be structured in different ways, but layering the network is important and must be considered when planning the network infrastructure.

Installing intrusion detection systems helps to monitor for attacks and parse audit logs that can use a large amount of storage and have a huge amount of information that a human cannot read and find a pattern for an attack at the same time it is happening. Intrusion detection systems help system and network administrators detect attacks and alert them about it while it shows the techniques in use to use possible breaches.

3.5.1 Encrypting your communication

There are several ways to move data into and out of a mainframe. Since the early 1970s, terminals connected to mainframes by using 3270 data streams. This high-performance protocol is still in use around the world and is how many developers connect to z/OS. By default, this data travels in clear text. Installations should evaluate the nature of the data that is transmitted over a 3270 connection and implement security measures, such as encryption, when warranted.

Enabling encrypted sessions requires configuration changes on both the host side and the client side. Fortunately, terminal emulators such as IBM PCOMM, IBM Host on Demand, and the open source x3270 emulators all support encryption of host sessions with simple configuration options.

Transport Layer Security/Secure Socket Layer

The Transport Layer Security/Secure Socket Layer (TLS/SSL) provides the processing capability that allows secure (encrypted) communication between two TCP/IP connection participants (one of which is a server or client application on the local z/VM host). Such communication may be secured by a static SSL connection or through Dynamic SSL/TLS, which allows a client or server application to control the acceptance and establishment of connections that are encrypted by using SSL.

For static TLS connections, no changes to a z/VM application server are necessary to participate in TLS. The application server does not perform any data encryption or decryption; this is handled by the z/VM TLS/SSL server.

Dynamic TLS connections are supported by the following z/VM TCP/IP application servers and clients, which were updated to accommodate this support:

- ▶ TCP/IP server
- ▶ SSL server
- ▶ FTP server
- ▶ FTP client
- ▶ Telnet server (Internal to the TCP/IP server)
- ▶ Telnet client
- ▶ SMTP server

Under the TLS protocol, the application server is always authenticated. To participate in a TLS session, an application server must provide a certificate to prove its identity. Server certificates are issued by Certifying Authorities (CAs), each of which establishes its own identity by providing a CA certificate. Server certificates and CA certificates are stored in a certificate database (also referred to as a *key* database) that is accessible to the TLS/SSL server.

Only TN3270 connections can perform client key exchanges, as shown in number 4 in Figure 3-2.

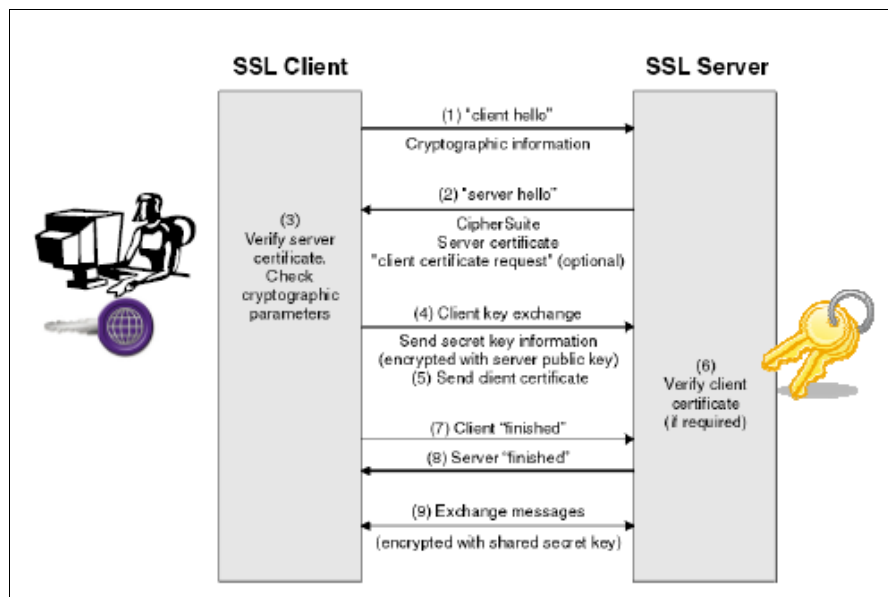


Figure 3-2 SSL scheme

You can configure the TLS/SSL server to meet industry and governmental cryptographic security requirements by updating the VMSSL keywords and parameters that are related to cipher suites and protocol levels. For z/VM V6.4 and onward support TLS 1.2, use this level of TLS/SSL for encrypting traffic to or within the hypervisor layer. Weaker cipher suites are disabled by default. If weaker encryption is required for compatibility purposes, it can be reenabled through the same keywords and parameters.

SMAPI ESM authorization support

With the PTF for APAR VM66167, SMAPI provides the following ESM interaction:

- ▶ When an ESM is present, programs can use the ESM for all SMAPI authorization decisions at the same granularity that is used with the SMAPI authorization mechanism. The ESM logs (or does not log) the decision that is based on its active policy, without SMAPI knowledge or intervention.
- ▶ When an ESM defers its authorization decision to SMAPI, one of the following actions is taken based on a configuration option:
 - The SMAPI authorization decision uses the authorization process. SMAPI calls the ESM to log the decision in the ESM-managed security log. SMAPI has no knowledge whether the ESM audit logging is enabled or disabled.
 - SMAPI treats the request as unauthorized.

Elliptic Curve Cryptography

With the PTF for APAR PI99184, the z/VM TLS/SSL server is enhanced to improve security through the enablement of Elliptic Curve Cryptography (ECC) cipher suites. ECC provides a faster, more secure mechanism for asymmetric encryption than standard RSA or DSS algorithms.

For more information (including PTF availability), see [this website](#).

Note: For more information about how to customize and enable encrypted communications to and from z/VM, see Chapter 4, “Installing and configuring z/VM”, in *The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3*, SG24-8147, and *Getting Started with Linux on Z Encryption for Data At-Rest*, SG24-8436.

3.5.2 z/VM Cryptographic definitions

When an LPAR is configured to benefit from hardware cryptography support, z/VM running in such an LPAR can use the hardware support for cryptographic operations to provide security to its guests. This section describes how to set up the z/VM definitions for guests running Linux on IBM LinuxONE.

By using the IBM LinuxONE cryptographic hardware, you gain security from using the CPACF and Crypto-Express6S through in-kernel cryptography APIs and for Linux on IBM Z, the libica cryptographic functions library. Using these features provides these benefits:

- ▶ File system encryption
- ▶ Communication encryption (to applications such as IBM HTTP Server)
- ▶ System security by providing advanced cryptographic functions

The way that z/VM provides this support is by granting access to the Adjunct Processor (AP) domains to the guests. From a system implementation perspective, an AP of a Crypto-Express5S feature is one of its internal cryptography engines (cryptography coprocessor units). AP designates to the processor, while AP ID specifies the number associated with it.

In a z/VM environment, it is expected that the LPAR running z/VM has access to multiple AP queues. There are two ways z/VM can provide access for the guests to the AP queues:

- ▶ Shared queue support (APVIRTual operand on the CRYPTO directory control statement)

Shared queue support provides for one or more Linux guest hardware encryption support for clear key operation. Clear key indicates that the key exists somewhere in the software stack in the clear. z/VM decides which AP queue is used.

- ▶ Dedicated queue support (APDEDicated operand on the CRYPTO directory control statement)

Dedicated queue support for a guest must be used if the guest needs secure key support and relies on stored encryption keys in the hardware coprocessors. Secure key support means that the key can never be found in a readable form outside the actual cryptographic hardware. For guests that use dedicated-queue support, z/VM does not intercept the AP instructions in the queue and instead allows the guest to run the AP instructions under SIE. In this case, no virtualization of AP queues is done.

When a key is defined in an IBM LinuxONE crypto environment as a secure key, the key is protected by another key that is called a *master key*. A clear key has not been encrypted under another key and has no additional protection within the cryptographic environment. For clear keys, the security of the keys is provided by operational procedures.¹

In an environment where the Linux guests on z/VM need only clear key support, use the shared queue support for hardware encryption. Even when z/VM virtualizes the AP queues for shared queue support, there must be at least one physical queue available for z/VM that is not dedicated to any guest.

Set up for a Linux guest to use cryptography cards

To enable a z/VM guest (Linux guest) to use the hardware cryptography support that is provided by the Crypto-Express6s feature, an entry must exist in the user directory of the Linux guest in the VM USER statement. This process is done by using the CRYPTO statement for each guest. For more information, see *CP Planning and Administration*, SC24-6271.

Guests with dedicated-queues support

For a Linux guest that needs access to dedicated-queues, the CRYPTO statement in the USER entry for the guest must contain which domain and which AP number is used, which means one or more AP queues are identified and reserved for this guest. There is no virtualization for these dedicated-queues, no sharing is done, and the queues are not available for other guests. With dedicated-queues, secure key and clear key operations can be performed by the Linux guest. The statement in the directory looks like the following one:

```
CRYPTO DOMAIN x APDED y
```

Where:

- ▶ DOMAIN x: x can be one or more domains that are defined for the z/VM LPAR.
- ▶ APDED y: y can be one or more APs (CEX6C cards) that are defined for the z/VM LPAR.

¹ For more information, see [Hardware Crypto for IBM Z & LinuxONE](#).

The combination of AP numbers and domain numbers should be unique across all cryptography users in the directory. Although you can use directory processing to specify the same AP and DOMAIN combination for multiple users, these users should not be logged on at the same time. If they are, more than one user might have concurrent access to the same AP queue. Directory processing does not enforce this restriction because duplicate definitions can be useful for backup configurations.

You can have multiple CRYPTO statements within one single user statement. However, if you choose different domains to different APs, all APs are available for all defined domains:

```
CRYPTO DOMAIN 10 APDED 1
CRYPTO DOMAIN 11 APDED 4
```

AP 1 and AP 4 are defined to the domains 10 and 12. It can also be shown as:

```
CRYPTO DOMAIN 10 11 APDED 1 4
```

The directory entry for the guests looks as shown in Example 3-8.

Example 3-8 Sample directory entries for dedicated-queues for cryptography access

```
USER EDI xxxxxxxx 64M 96M ABCDEFG
  INCLUDE IBMDFLT
  CRYPTO DOMAIN 004 APDED 005
  CRYPTO DOMAIN 3 APDED 0 7
  IPL CMS PARM FILEPOOL VMSYSU AUTOOCR
  OPTION LNKNOPAS QUICKDSP
  MDISK 0191 3390 71 10 ZVMUSR MR
```

The privileged command, **Q CRYPTO DOMAIN USERS** shows the output that is shown in Example 3-9.

Example 3-9 Result of a Q CRYPTO DOMAIN command

```
q crypto dom users
AP 000 CEX6C Domain 002 available shared unspecified
AP 000 CEX6C Domain 003 available dedicated to EDI dedication
AP 000 CEX6C Domain 004 available dedicated to EDI dedication
AP 002 CEX6C Domain 002 available shared unspecified
AP 002 CEX6C Domain 003 available free unspecified
AP 002 CEX6C Domain 004 available free unspecified
AP 003 CEX6C Domain 002 available free unspecified
AP 003 CEX6C Domain 003 available free unspecified
AP 003 CEX6C Domain 004 available free unspecified
AP 005 CEX6C Domain 002 available free unspecified
AP 005 CEX6C Domain 003 available dedicated to EDI dedication
AP 005 CEX6C Domain 004 available dedicated to EDI dedication
AP 006 CEX6C Domain 002 available free unspecified
AP 006 CEX6C Domain 003 available free unspecified
AP 006 CEX6C Domain 004 available free unspecified
AP 007 CEX6C Domain 002 available free unspecified
AP 007 CEX6C Domain 003 available dedicated to EDI dedication
AP 007 CEX6C Domain 004 available dedicated to EDI dedication
```

Guests with shared-queue support

For a Linux guest that needs access to clear key cryptography operations, shared access to AP queues is the preferred way to implement this access. For this case, the CRYPTO statement in the USER entry for the guest needs to indicate that access to virtual queues is wanted. No domain and no AP queue must be specified. The Linux guest gets one virtualized card and one random virtual queue on one random virtual AP. The AP number and domain are chosen by z/VM and are not identical to the one for the z/VM LPAR.

Note: As of IBM LinuxONE, you can now specify a CRYPTO APVIRT statement in your System Configuration file. This specification allows the system administrator to designate specific AP domains that are attached to the LPAR as “Reserved for APVIRT”.

For this support, z/VM uses all available AP queues, which are not dedicated to other guests, and these are shared between all guests that use the shared support. If there are multiple AP types available for z/VM, then z/VM chooses the best AP type for acceleration for the Linux guest. When a type is selected, z/VM routes all cryptography requests from the guest to however many queues or cards of that type are available. The statement in the directory looks like the following example:

CRYPTO APVIRT

The AP queues number and the domain number, which are provided by z/VM to these two guests, are virtual numbers and do not correspond to the “real” domains and APs, which are used by z/VM to run the cryptography requests of these guests. The directory entry for the guests looks like what is shown in Example 3-10.

Example 3-10 Directory entry with dedicated and shared cryptography queues

```
USER GUESTL1 xxxxxx 256M 1G G
  INCLUDE IBMDFLT
  IPL CMS
  MACH XA
  NICDEF C200 TYPE QDIO LAN SYSTEM VSWITCH1
  CRYPTO DOMAIN 9 APDED 2 3
----- 3 line(s) not displayed -----
USER GUESTL2 xxxxxx 256M 1G G
  INCLUDE IBMDFLT
  IPL CMS
  MACH XA
  NICDEF C200 TYPE QDIO LAN SYSTEM VSWITCH1
  CRYPTO APVIRT
```

To update the USER entry in the directory to contain the CRYPTO statement, you can use an editor and change all necessary USER entries. In an environment with DirMaint, proceed as described next to provide shared access to a Linux guest LNXSU1 for clear key operation and dedicated access to the AP queue with domain 11 and AP number 02 to LNXSU2 for secure key operation.

To change the directory for EDI to get shared access to the cryptography hardware, run the command **dirm for EDI crypto**. The pane that is shown in Figure 3-3 on page 48 opens. In this DirMaint pane, select APVIRTUAL (for the operand APVIRT in the CRYPTO statement) with any character and press F5 to submit the request.

Example 3-11 shows the response to the **QUERY CRYPTO** command in z/VM environment where cryptographic units are available but no guests are assigned access.

Example 3-11 Crypto units available: z/VM guests do not have access to AP queues

CP Q CRYPTO

Crypto Adjunct Processor Instructions are installed

Using also the AP operand, you get more information about the installed AP queues, as shown in Example 3-12. In this example, domains are available for shared access (clear key) of the queues.

Example 3-12 Response to QUERY CRYPTO

q crypto ap

AP 000 CEX6C	Domain 002 available	shared	unspecified
AP 000 CEX6C	Domain 003 available	free	unspecified
AP 000 CEX6C	Domain 004 available	free	unspecified
AP 002 CEX6C	Domain 002 available	shared	unspecified
AP 002 CEX6C	Domain 003 available	free	unspecified
AP 002 CEX6C	Domain 004 available	free	unspecified
AP 003 CEX6C	Domain 002 available	free	unspecified
AP 003 CEX6C	Domain 003 available	free	unspecified
AP 003 CEX6C	Domain 004 available	free	unspecified
AP 005 CEX6C	Domain 002 available	free	unspecified
AP 005 CEX6C	Domain 003 available	free	unspecified
AP 005 CEX6C	Domain 004 available	free	unspecified
AP 006 CEX6C	Domain 002 available	free	unspecified
AP 006 CEX6C	Domain 003 available	free	unspecified
AP 006 CEX6C	Domain 004 available	free	unspecified
AP 007 CEX6C	Domain 002 available	free	unspecified
AP 007 CEX6C	Domain 003 available	free	unspecified
AP 007 CEX6C	Domain 004 available	free	unspecified

QUERY VIRTUAL CRYPTO command

The **QUERY VIRTUAL CRYPTO** command shows status information about the virtual cryptographic facilities of the z/VM guest. If the guest to which you are currently logged in to does not have access to cryptography queues, the response in Example 3-13 is shown.

Example 3-13 Guest does not have access to cryptography queues

CP Q V CRYPTO

No AP Crypto Domains are available

Trusted Key Entry support

The Trusted Key Entry (TKE) feature is a combination of workstation hardware and TKE Licensed Internal Code. It provides key management functions for operating systems, such as Linux on IBM LinuxONE and z/OS. Crypto-Express Cards are used to provide hardware support to most cryptographic functions through the Cryptographic Coprocessor Feature (CCF).

TKE provides a secure, remote, and flexible method for providing Master Key Part Entry, and to manage remotely PCIe cryptographic coprocessors for the crypto domains, that is, through smart cards or other secured devices. The cryptographic functions on the TKE are run by one PCIe cryptographic coprocessor.

The communication between the TKE workstation and IBM LinuxONE servers occurs through a TCP/IP connection, which is available through Ethernet LAN connectivity only.

Note: For more information about how to set up the TKE and managing master keys for crypto domains, see *TKE Workstations User's Guide*, SC14-7511.

To enable guests running under z/VM benefit from cryptographic hardware, crypto domains, and crypto coprocessors must be attached to z/VM LPARS through HMC dialogues. After this process is done, crypto domains can be dedicated to or shared with other LPARs.

3.6 z/VM connectivity

Connectivity in z/VM can be provided by customizing TCP/IP. The TCP/IP VM provides the primary TCP/IP service that is called the *stack*. The stack controls the network interfaces, such as Open Systems Adapter (OSA), and supports the application programming interfaces (APIs).

For more information about how to set up and define the stack, see Chapter 19, “Configuring the TCP/IP Server”, in *TCP/IP Planning and Customization*, SC24-6331.

3.6.1 DEVICE and LINK statements

For TCP/IP to use network devices on z/VM, you must ensure that the device addresses are attached to the TCP/IP VM by doing one of the following actions:

- ▶ Modify the DTCPARMS file to enable the necessary devices to be attached by using the :Attach. tag.
- ▶ Add the appropriate DEDICATE control statements to the TCP/IP VM's directory entry.

z/VM does not require a device definition in the system configuration file or HCPRIO. The device attributes are determined automatically during device initialization.

Example 3-14 shows the Device and Link configuration statements for an OSA device that is found in the PROFILE TCPIP file.

Note: Although system configuration is not required from an I/O standpoint, you might need to code RDEVICE statements in SYSTEM CONFIG to set up the Equivalence ID (EQID) for the QDIO devices in an SSI environment.

Example 3-14 DEVICE and LINK statements for an OSA device in the PROFILE TCPIP

```
DEVICE DEV$04B0 OSD 04B0  
LINK OSA01 QDIOETHERNET DEV$04B0 PATHMTU 1500 VLAN 20
```

VLAN

In z/VM environment, the virtual servers that are connected to each other through a virtual LAN (VLAN) which eliminates the requirement for any physical cabling or external networking connection among them. Functioning as an internal LAN, it moves data at memory speed between the virtual servers with high throughput and low latency communication path.

VSWITCH

The z/VM Virtual switch (VSWITCH) is built on guest LAN technology and consists of a network of virtual adapters that can be used to interconnect guest systems (see Figure 3-5). The virtual switch can also be associated with one or more IBM Open Systems Adapter-Express (OSA) ports. This capability allows access to external LAN segments without requiring an intermediate router between the external LAN and the internal z/VM guest LAN.

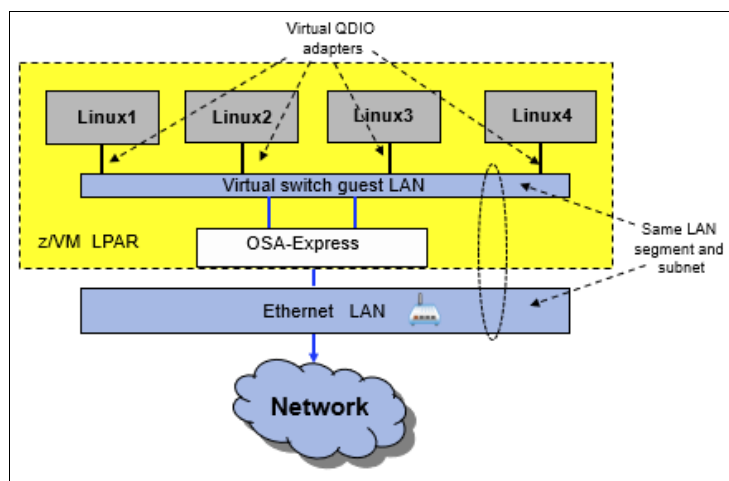


Figure 3-5 z/VM Virtual switch (VSWITCH)

HiperSockets

IBM HiperSockets are an extension to the queued direct I/O (QDIO) Hardware Facility, providing a microcode only, low latency communications vehicle for Internet Protocol (IP) inter-program communication (IPC). With the use of HiperSockets, a program can directly communicate with a program running within the same LPAR and across any LPAR within the same central processor complex.

3.6.2 HiperSockets VSWITCH Bridge

Another type of connectivity is available with the IBM LinuxONE using z/VM V7.1. The HiperSockets VSWITCH Bridge was introduced to allow an internal HiperSockets network to be extended outside the IBM LinuxONE processor complex. By using this capability, a network configuration can be greatly simplified.

Consider the following points:

- ▶ Guests of z/VM that need access to VSWITCH and HiperSockets hosts must be attached to only connection types to have access to both, without routing.
- ▶ HiperSockets networks in different IBM LinuxONE CPCs can be logically connected to each other, which means that z/VM guests that use LGR can treat the HiperSockets networks in different CPCs as the same because traffic is bridged between them.

The HiperSockets VSWITCH Bridge does not require a TCP/IP stack to function. The capability is provided by the z/VM CP system service that supports VSWITCH operation. The bridge is set up by defining a HiperSockets connection as an UPLINK port on a VSWITCH. The HiperSockets VSWITCH Bridge takes care of all issues relating to moving a frame from QDIO OSA frame type to IQDIO HiperSockets mode (and vice versa).

Note: For more information, see *IBM HiperSockets Implementation Guide*, SG24-6816.

3.6.3 Security considerations

Because HiperSockets often are an isolated network with no exposure outside the IBM LinuxONE CPC, security considerations exist for the use of the HiperSockets VSWITCH Bridge. Bridging a HiperSockets CHPID that is used for secure cross-system traffic within an IBM LinuxONE CPC to an external network connection might be a cause of concern that the secure traffic may be exposed. However, because HiperSockets are a LAN segment, security considerations exist whether they are transmitting data in-memory or over OSA CHPIDs.

Note: The HiperSockets-VSWITCH Bridge is not a promiscuous bridge; it does not passively transfer all traffic appearing on the HiperSockets onto the VSWITCH or vice versa. It actively sends only frames with destinations that are unknown on the HiperSockets to the VSWITCH. Likewise, the VSWITCH sends only frames to the HiperSockets for addresses that the HiperSockets registered to the VSWITCH.

Because of the way the function works, it is not feasible that secure traffic on a HiperSockets CHPID might be exposed by the HiperSockets-VSWITCH Bridge. Consider the following points:

- ▶ The function copies frames only to destinations that are unknown on the HiperSockets over to the VSWITCH.
- ▶ The HiperSockets network traffic analyzer (NTA) function, which is the only method that is available for tracing or 'sniffing' traffic on a HiperSockets, only functions with Linux running natively in the LPAR authorized for NTA.

The number of HiperSockets networks that are available in an IBM LinuxONE CPC makes it possible that a dedicated HiperSockets virtual network might be used for the HiperSockets VSWITCH Bridge function without any perceived risk to the traffic that is carried on HiperSockets that are in use.

3.7 Remote Spooling Communications Subsystem

Remote Spooling Communications Subsystem (RSCS) Networking for z/VM is a networking program that enables users on a z/VM system to send messages, files, commands, and jobs to other users within a network. It is an easy way to transfer data files (as spool files) among z/VM systems or other systems, such as, z/OS. It also acts as a print server for remote printers that are attached to other z/VM systems or a Internet Protocol network. Through RSCS, users can send and receive messages, files, issue commands, and print and submit jobs within their networks.

RSCS can communicate with system nodes that are running under the control of network job entry (NJE) compatible subsystems, such as:

- ▶ JES2 or JES3
- ▶ RSCS
- ▶ VSE/POWER
- ▶ IBM AS/400® Communications Utilities
- ▶ Products that provide NJE functions for Linux or IBM AIX®

NJE is the native peer-to-peer networking protocol for IBM mainframes running RSCS. It is flexible so that you can customize it to meet the changing needs of your installation and network. Using exit facilities and control files, such as the configuration file and events file, you can set up and tailor the way RSCS works and establish security rules by using specific exits.

Since the earlier z/VM versions, the encryption support of TCPNJE traffic was implemented. A TCPNJE link connects the local RSCS system to a remote NJE system through TCP/IP. Because TCPNJE uses the z/VM TCP/IP stack, this configuration also supports encryption by using the TLS/SSL server, as described in 3.5.1 “Encrypting your communication” on page 42.

For more information about RSCS and the NJE communication protocol, see [Best Practices for NJE on z/VM: Security Configuration Steps for RSCS and VMBATCH](#).

Main functions

RSCS is a subsystem that can perform the following functions:

- ▶ Handle data being sent to, from, or through z/VM systems.
- ▶ Store and retrieve input and output data files on the z/VM system spool.
- ▶ Use communications equipment to transfer data between its z/VM system and remote users, devices, and other systems.

How RSCS fits into your installation

RSCS uses the z/VM system spool to manage file transfers. It uses the spool for temporary file storage and to store files being transferred between its local system and remote users, devices, or systems.

To manage files that are spooled to remote nodes, RSCS relies on tag information. A spool file tag becomes part of each data file that is spooled to RSCS. The tag contains information that describes where the file came from (origin information) and where it is going (destination information).

RSCS EXITS

RSCS exits are used to restrict the sending and receiving files of programs that can affect the performance of the system, such as executable files that can have malware. Using exit facilities and control files, such as the configuration file and EVENTS file, you can set up and tailor the network where RSCS functions. You can use the exit facility to modify RSCS processing to meet any special functional requirements for your installation.

Note: For more information, see *RSCS Networking Exit Customization*, SC24-6317.

Security aspects

The Gateway Security Modification (GSM) is a feature of RSCS that can reject any files that go through RSCS. You can create control files to restrict certain users from sending files to a specific system and issue monitor commands of RSCS.

As a preferred practice, create a rule so that the users have permission to send/receive files to other systems while restricting the areas that they can use.

Table 3-1 lists the main security packages that must be implemented in your environment.

Table 3-1 Some of RSCS Exits focusing some security aspects

Package name	Main function	Exit
Gateway security modifications (GSM)	Controls the data traffic.	Gateway programming interface, Exit 0, Exit 1, Exit 14, Exit 15, Exit 19, Exit 21, Exit 29, and Exit 32
Selective file filter (SFF)	Purges unwanted files.	Exit 0, Exit 1, Exit 15, Exit 21, and Exit 29
Simple security package (SSS)	Limits file traffic or user ID usage.	Exit 0, Exit 1, Exit 14, Exit 15, Exit 19, Exit 21, and Exit 32

RSCS advantages

In a multisystem network, because the systems are interconnected, data can be moved through and between them from any system and to any system. RSCS running under z/VM (in addition to what it can do in a single-system environment) can do networking.

To users, networking means they can perform the following tasks:

- ▶ Exchange data with users on the same system.
- ▶ Exchange data with systems and users at other locations.
- ▶ Send jobs to other systems for processing.
- ▶ Direct processed output to devices, such as printers and punches, that are connected to another system.

Employees can get that data that they need from other systems. When work passes from one phase to another, moving as it does from department to department, RSCS can transfer data from one employee to the next, from one system to another system. Employees can do the following actions:

- ▶ Correspond electronically.
- ▶ Use programs on another system to process their jobs.
- ▶ Send jobs from a remote workstation at their location to the local or to a remote system.
- ▶ Direct output to RSCS-controlled 3270 printers or ASCII devices from jobs they submitted to either local or remote systems.
- ▶ Send or receive output for other system-controlled printers through RSCS.
- ▶ Send an output file that they created for another employee to print on that employee's system.

You can buy and locate resources (processors, computer programs, and I/O devices) to fit your business needs, whether by departments or regions. Because these resources can be shared, you can distribute the workload of your business and improve your employees access to these resources. This can lead to greater efficiency and productivity in your business.

Note: For more information, see *RSCS Networking Planning and Configuration*, SC24-6320.



IBM Resource Access Control Facility Security Server for IBM z/VM

The IBM Resource Access Control Facility (RACF) Security Server 710 for z/VM 7.1.0, builds on the function that is provided by previous releases. Significant System Programming Enhancements (SPEs) were made since September 2015 to increase the capability and security of RACF.

This chapter describes the processes of installing, configuring, managing, monitoring, auditing, and controlling of RACF resources. This chapter follows the concepts that are presented in *Secure Configuration Guide for z/VM*, SC24-6323.

Note: *Secure Configuration Guide for z/VM*, SC24-6323, describes installing and customizing z/VM and RACF in a way that meets requirements for certification for the Common Criteria Operating System Protection Profile (OSPP). This chapter does not describe all of the steps that are involved in setting up the certified configuration. However, important basic steps are outlined in *Secure Configuration Guide for z/VM*, SC24-6323 that are important for a normal RACF installation.

A Directory Manager is recommended for a Single System Image (SSI) environment to maintain synchronization between object directories. Although the Common Criteria evaluation does not make any claims about z/VM, security administrators should determine what Directory Manager best fits their security policies and act accordingly.

This chapter assumes the use of the IBM z/VM Directory Manager (DirMaint) product for managing the user directory of the system, with RACF and DirMaint configured to work together. Unlike other operating systems, z/VM separates the processes of user definition and security administration.

You must first define the virtual machine (VM) in the user directory, which provides basic resource control configuration. If you are using an external security manager (ESM), you must also add users and define user resources profile to the ESM database.

For more information about DirMaint installation and its use, see Appendix A, “DirMaint implementation” in *Security on z/VM*, SG24-7471, and in *IBM Virtualization Cookbook z/VM* 6.3, SG24-8147.

For more information about the DirMaint-RACF Connector, see [this website](#).

This chapter includes the following topics:

- ▶ 4.1, “RACF z/VM concepts” on page 57
- ▶ 4.2, “Activating and configuring RACF” on page 59
- ▶ 4.3, “RACF management processes” on page 88

4.1 RACF z/VM concepts

This section describes some concepts about RACF to protect the security of a z/VM installation.

4.1.1 External security manager

An external security manager (ESM) is software that provides enhanced security access control over the functions that are provided by the operating system. As with z/OS®, z/VM implements the ESM concept so that you can choose and configure a security manager that fits your organization needs. RACF Security Server for z/VM is the IBM ESM for the z/VM platform.

The ESM receives requests from resource managers on the system (CP, Shared File System, TCP/IP, FTP, and so on) on behalf of userids (VMs) that must access resources. When the resource manager is enabled for an ESM, it calls the ESM to check whether the z/VM userid (VM) has the proper authorization to access the resource.

The ESM performs several operations to determine what happens next. It first checks to see whether it is set up to be responsible for the type of resources being requested (for example, minidisks or virtual switches). If the ESM is responsible, it checks whether the z/VM userid (VM) has direct access to the resource, or is a member of a group that has access to the resource. If the z/VM userid (VM) or a group of which that userid is a member has the authority, access is granted to the resource.

The ESM can also be configured to control what occurs when the ESM is not responsible for the type of resources being requested, or an explicit resource definition is not available to control access to the particular resource being requested. The ESM can be configured to deny access to the resource in that situation.

Alternatively, the ESM can defer authorization to z/VM CP component, which means that the resource manager then must use the traditional access methods (for example, passwords that are configured in the user directory in the case of minidisks) to control access to the resource.

Note: The default action that is taken by RACF is to defer to CP. For more information about this action and how to change this default action, see 4.2.5, “Using HCPRWAC” on page 85.

The z/VM resource managers interface with the ESM by using the RPI interface.

4.1.2 Security policy

An ESM must be configured to support its role in maintaining system security and integrity. In the case of RACF, this configuration occurs in the following areas:

- ▶ RACF options
- ▶ Classes
- ▶ User definitions
- ▶ Resource definitions (profiles)
- ▶ Access control lists (ACLs)
- ▶ Audit settings

The combination of these configuration settings must follow your organization's *security policy*. The policy is agreed to across operational and business areas in your organization, and covers issues, such as the following examples:

- ▶ The default levels of access that should exist for different types of resources.
- ▶ Whether access to resources is managed by grouping them or by maintaining separate ACLs for each resource.
- ▶ What level of tracking of access requests occur (for example, auditing all access requests or just failures).
- ▶ The roles and responsibilities of administrators and users in the organization, including the separation of duties between those roles.

The security policy is implemented by using the configurations and settings in RACF.

Note: For more information about how to start implementing a security policy by using RACF, see Chapter 2, "Organizing for RACF Implementation" of *RACF Security Server Security Administrator's Guide*, SC24-6311.

The "default" security policy

The process that is described in this chapter is based on the steps that are outlined in *Program Directory for RACF Security Server for z/VM*, GI13-4358. In this process, you use a utility that is supplied with RACF to create an initial database of RACF profiles and ACLs that are based on your system's current CP directory. In effect, this process implements a "default" security policy. Likewise, the functions of other system components, such as the DirMaint-RACF connector.

The policy that is inferred by using the basic RACF utilities is sufficient for most installations. After all, it is based on the policy as implemented by the CP directory, and most installations do not change the definitions that are contained there. The following basic rules are inherent in this policy:

- ▶ All resources include profiles that protect that resource specifically (known as *discrete profiles*).
- ▶ The owner of a resource has full authority over that resource, including the authority to grant other users access to it.
- ▶ Administration roles (auditor and security administrator) are separated.

Some highly sensitive organizations, or those installations with experienced security administration staff, might want to adjust the output that is generated by the utilities so that they better reflect the specific needs of the organization.

Optimizing administration

Another aspect of the "default" operation of RACF functions and utilities is that the number of resource profiles and ACLs is not optimized. With every resource having a discrete profile, the number of profiles in the database can grow, and it can become more complex to manage many resources. For more information about a more streamlined way to manage resource protection, see 4.3.4, "Securing your minidisks with RACF" on page 97.

If you use only the RACF utilities and tools for management, such as the DirMaint-RACF connector, this issue is insignificant here. Although the number of profiles in RACF might become large, the utilities keep them up to date.

If you choose to optimize your RACF operation and use techniques, such as generic profiles and group-based access control, be aware that you might have to do some work to implement your own system to manage your altered policy. For example, you might not use the DirMaint-RACF connector to manage minidisk profiles (the VMMDISK class) if you use group membership to authorize minidisk access.

4.2 Activating and configuring RACF

RACF for z/VM is included with the z/VM 7.1.0 system deliverable and managed by using Virtual Machine Serviceability Enhancement Staged/Extended (VMSES/E). RACF for z/VM is a priced product that is supplied in a disabled state. It must be enabled and configured by the system programmer before you use it.

The Program Directory for the product describes the installation process and can be downloaded from [this website](#).

Note: Make sure that your installation includes a license for RACF before you activate it.

For more information about the step-by-step process to enable and configure the product, see *Program Directory for RACF Security Server for z/VM*, GI13-4358.

This process includes the following main steps:

1. Set RACF to the ENABLED state by using the VMSES/E tools.
2. Perform post activation steps, as described in 4.2.1, “Post-activation tasks” on page 59.
3. Build the RACF enabled CPLOAD MODULE, as described in 4.2.2, “Building the RACF enabled CPLOAD MODULE” on page 77.
4. Update the RACF database and options, as described in 4.2.3, “Updating the RACF database and options” on page 80.
5. Place RACF into production, as described in 4.2.4, “Placing RACF into production” on page 84.
6. Update the authorization process, as described in 4.2.5, “Using HCPRWAC” on page 85.

Print the procedural checklist from the RACF Security Server for z/VM Program Directory so that you do not miss any important steps in the process.

This chapter assumes that the reader has basic z/VM system programming knowledge. Experience with VMSES/E and its processes is helpful, but not essential.

4.2.1 Post-activation tasks

This section describes the following tasks that you must perform after you activated the RACF code. These tasks reflect preferred practices that were tested in the ITSO test environment:

- ▶ Allocating the RACF DASD
- ▶ Defining RACF user IDs
- ▶ Evaluating the minidisk access
- ▶ Updating the RACF user ID directory entry
- ▶ Running RPIDIRCT
- ▶ Customizing the processing of SMF records
- ▶ Password encryption algorithm
- ▶ Removing ICHRCX02

Allocating the RACF DASD

The default definitions of the minidisks that are used to hold the primary and backup RACF database are not recommended for production use. By default, both minidisks are defined on the same volume, which means the database might be lost if that volume is lost. Also, if you enabled the z/VM SSI feature, you are required to have the RACF database on full-pack minidisks rather than the default minidisks.

If you are not using SSI, move the RACFVM 300 disk to a different volume on your system. This process ensures that if the volume that is holding the 200 disk is damaged, you do not lose all the RACF data. Then, you can proceed with “Defining RACF user IDs” on page 66.

If you are using SSI, follow the directions in the following sections (from the “Sharing RACF Databases in a z/VM Single System Image Cluster” section of *z/VM: RACF Security Server System Programmer’s Guide*, SC24-6312) to define both RACF database minidisks as full-pack minidisks:

- ▶ Moving the RACFVM 200 and 300 minidisks
- ▶ Creating definitions for RACFVM 200 and 300 disks as full-pack minidisks
- ▶ Defining the RACF database disks as shared
- ▶ Defining the initial RACF database

Moving the RACFVM 200 and 300 minidisks

Remove the minidisk definitions. Example 4-1 shows the use of the **DIRM CHVADDR** command to move the definitions to a different device address.

Note: In this example, we decided not to delete these minidisks, but rather to move them to a different address so that we can use them as the source for copying the supplied initialized primary and backup databases later in the process. This action also provides a number of disks across the system that can be used as destinations for backing up the databases. Backing up the RACF database is described in 4.3.7, “Backing up the RACF database” on page 103.

Example 4-1 Moving the minidisk definitions

dirm for racfvm-1 chvaddr 200 to f200

```
DVHXMT1191I Your CHVADDR request has been sent for processing to
DVHXMT1191I DIRMAINT at ITS0ZVM1.
Ready; T=0.01/0.01 16:57:43
DVHREQ2288I Your CHVADDR request for RACFVM-1 at * has been accepted.
DVHBIU3450I The source for directory entry RACFVM-1 has been updated.
DVHBIU3424I The next ONLINE will take place immediately.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHBIU3428I Changes made to directory entry RACFVM-1 have been placed
DVHBIU3428I online.
DVHREQ2289I Your CHVADDR request for RACFVM-1 at * has completed; with
DVHREQ2289I RC = 0.
```

dirm for racfvm-1 chvaddr 300 to f300

```
DVHXMT1191I Your CHVADDR request has been sent for processing to
DVHXMT1191I DIRMAINT at ITS0ZVM1.
Ready; T=0.01/0.01 16:57:52
DVHREQ2288I Your CHVADDR request for RACFVM-1 at * has been accepted.
DVHBIU3450I The source for directory entry RACFVM-1 has been updated.
DVHBIU3424I The next ONLINE will take place immediately.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
```

```
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHBIU3428I Changes made to directory entry RACFVM-1 have been placed
DVHBIU3428I online.
DVHREQ2289I Your CHVADDR request for RACFVM-1 at * has completed; with
DVHREQ2289I RC = 0.
```

Example 4-1 on page 60 shows the first member of the SSI cluster. We repeated the commands for each of the other members of the cluster (IDs RACFVM-2, RACFVM-3, and RACFVM-4).

Creating definitions for RACFVM 200 and 300 disks as full-pack minidisks

Next, add the new full pack minidisk definitions to the RACFVM user. In this example, we obtain two 3390 volumes that we can use as RACF database volumes and add them to the RACFVM user. In this example, we add the disks to the directory IDENTITY entry for RACFVM, which is recommended when all members of the SSI cluster see the disks at the same device address.

Note: If your configuration does not have symmetric device addressing (that is, the RACF database disks do not have the same device address across all members of the SSI cluster), you must add the disks to each SUBCONFIG entry by using the appropriate device addresses.

Example 4-2 shows how we achieved this task by using the **DIRM AMDISK** command in DirMaint.

Example 4-2 Add disk devices to RACFVM

```
dirm for racfvm amdisk 200 3390 devno 3b07 mwv pws read write multiple
DVHXMT1191I Your AMDISK request has been sent for processing to DIRMAINT
DVHXMT1191I at ITS0ZVM1.
Ready; T=0.01/0.01 16:40:19
DVHREQ2288I Your AMDISK request for RACFVM at * has been accepted.
DVHSCU3541I Work unit 10164020 has been built and queued for processing.
DVHSHN3541I Processing work unit 10164020 as VIC from ITS0ZVM1,
DVHSHN3541I notifying VIC at ITS0ZVM1, request 11 for RACFVM SSI node *;
DVHSHN3541I to: AMDISK 0200 3390 DEVNO 3B07 MWV PWS XXXX XXXXX XXXXXXXX
DVHBIU3450I The source for directory entry RACFVM has been updated.
DVHBIU3424I The next ONLINE will take place immediately.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHBIU3428I Changes made to directory entry RACFVM have been placed
DVHBIU3428I online.
DVHSHN3430I AMDISK operation for RACFVM address 0200 has finished
DVHSHN3430I (WUCF 10164020).
DVHREQ2289I Your AMDISK request for RACFVM at * has completed; with RC =
DVHREQ2289I 0.
dirm for racfvm amdisk 300 3390 devno 3c07 mwv pws read write multiple
DVHXMT1191I Your AMDISK request has been sent for processing to DIRMAINT
DVHXMT1191I at ITS0ZVM1.
Ready; T=0.01/0.01 16:44:41
DVHREQ2288I Your AMDISK request for RACFVM at * has been accepted.
DVHSCU3541I Work unit 10164442 has been built and queued for processing.
DVHSHN3541I Processing work unit 10164442 as VIC from ITS0ZVM1,
```

```

DVHSHN3541I notifying VIC at ITS0ZVM1, request 12 for RACFVM SSI node *;
DVHSHN3541I to: AMDISK 0300 3390 DEVNO 3C07 MWV PWS XXXX XXXXX XXXXXXXX
DVHBIU3450I The source for directory entry RACFVM has been updated.
DVHBIU3424I The next ONLINE will take place immediately.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHBIU3428I Changes made to directory entry RACFVM have been placed
DVHBIU3428I online.
DVHSHN3430I AMDISK operation for RACFVM address 0300 has finished
DVHSHN3430I (WUCF 10164442).
DVHREQ2289I Your AMDISK request for RACFVM at * has completed; with RC =
DVHREQ2289I 0.
DVHREQ2288I Your LINK request for RACMNT-1 at * has been accepted.
DVHBIU3450I The source for directory entry RACMNT-1 has been updated.
DVHBIU3426I Object directory updates are currently disabled.
DVHREQ2289I Your LINK request for RACMNT-1 at * has completed; with RC =
DVHREQ2289I 0.
. . .
< above output repeats for RACMNT-2, RACMNT-3, and RACMNT-4 >
. . .
DVHREQ2288I Your ONLINE request for VIC at * has been accepted.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHREQ2289I Your ONLINE request for VIC at * has completed; with RC = 0.

```

Note: If you use IBM Geographically Dispersed Parallel Sysplex (GDPS), you cannot use the **DEVNO** parameter on MDISK statements. Instead, you *must* follow the instructions that are contained in the Program Directory for defining the RACFVM 200 and 300 disks.

The RACMAINT user links to RACFVM 200 and 300 disks. To make sure that RACMAINT has correct access to the disks, these LINK definitions must be updated. In this example, we made these updates by creating a DirMaint batch file, as shown in Figure 4-1 on page 63.

```

RACMLINK DIRMBAT A1 F 80 Trunc=80 Size=18 Line=0 Col=1 Alt=0

00000 * * * Top of File * * *
00001 offline
00002 for racmnt-1 link racfvm 200 200 delete
00003 for racmnt-1 link racfvm 300 300 delete
00004 for racmnt-1 link racfvm 200 200 mw
00005 for racmnt-1 link racfvm 300 300 mw
00006 for racmnt-2 link racfvm 200 200 delete
00007 for racmnt-2 link racfvm 300 300 delete
00008 for racmnt-2 link racfvm 200 200 mw
00009 for racmnt-2 link racfvm 300 300 mw
00010 for racmnt-3 link racfvm 200 200 delete
00011 for racmnt-3 link racfvm 300 300 delete
00012 for racmnt-3 link racfvm 200 200 mw
00013 for racmnt-3 link racfvm 300 300 mw
00014 for racmnt-4 link racfvm 200 200 delete
00015 for racmnt-4 link racfvm 300 300 delete
00016 for racmnt-4 link racfvm 200 200 mw
00017 for racmnt-4 link racfvm 300 300 mw
00018 online immed
00019 * * * End of File * * *

```

Figure 4-1 RACMLINK DIRMBAT for batch update of RACMAINT links

Running the RACMLINK DIRMBAT file by using the **DIRM BATCH RACMLINK DIRMBAT** command that produced the output that is shown in Example 4-3.

Example 4-3 Update the RACMAINT links to RACF database disks

dirm batch racmlink dirmbat

```

PUN FILE 0077 SENT TO DIRMAINT RDR AS 4617 RECS 0026 CPY 001 0 NOHOLD NOKEEP
DVHXMT1191I Your BATCH request has been sent for processing to
DVHXMT1191I DIRMAINT at ITS0ZVM1.
Ready; T=0.01/0.01 13:40:17
DVHREQ2288I Your BATCH request for VIC at * has been accepted.
DVHREQ2289I Your BATCH request for VIC at * has completed; with RC = 0.
DVHREQ2288I Your OFFLINE request for VIC at * has been accepted.
DVHREQ2289I Your OFFLINE request for VIC at * has completed; with RC =
DVHREQ2289I 0.
DVHREQ2288I Your LINK request for RACMNT-1 at * has been accepted.
DVHBIU3450I The source for directory entry RACMNT-1 has been updated.
DVHBIU3426I Object directory updates are currently disabled.
DVHREQ2289I Your LINK request for RACMNT-1 at * has completed; with RC =
DVHREQ2289I 0.
DVHREQ2288I Your LINK request for RACMNT-1 at * has been accepted.
DVHBIU3450I The source for directory entry RACMNT-1 has been updated.
DVHBIU3426I Object directory updates are currently disabled.
DVHREQ2289I Your LINK request for RACMNT-1 at * has completed; with RC =
DVHREQ2289I 0.
DVHREQ2288I Your LINK request for RACMNT-1 at * has been accepted.
DVHBIU3450I The source for directory entry RACMNT-1 has been updated.
DVHBIU3426I Object directory updates are currently disabled.

```

Note: The batch file is not the only way to complete this task. You can issue each of the commands in the batch file separately, or you can use **DIRM GET** command to retrieve each of the RACMAINT SUBCONFIG entries, edit the files directly, and use the **DIRM REPLACE** command to update them.

To verify that the directory update is correct, run **DIRM REVIEW** for RACFVM. The output is shown in Example 4-4 (for clarity, only the statements belonging to the first member of the SSI cluster are shown).

Example 4-4 DIRM REVIEW for RACFVM after minidisk updates

```
* * * Top of File * * *
IDENTITY RACFVM XXXXXXXX 20M 20M ABCDEGH
DVHRXV3366I The following configurations will be used on SSI nodes.
DVHRXV3366I The following configuration RACFVM-1 will be used on SSI
DVHRXV3366I node ITS0ZVM1.
SUBCONFIG RACFVM-1
  LINK MAINT 0190 0190 RR * CMS SYSTEM DISK
  LINK MAINT 019D 019D RR * HELP DISK
  LINK MAINT 019E 019E RR * PRODUCT CODE DISK
  MDISK 0191 3390 8235 009 ZA1RS1 MR XXXXXXXX XXXXXXXX XXXXXXXX
  MDISK F200 3390 8218 017 ZA1RS1 MW XXXXXXXX XXXXXXXX XXXXXXXX
  MDISK 0490 3390 8244 070 ZA1RS1 MR XXXXXXXX XXXXXXXX XXXXXXXX
  MDISK 0305 3390 8314 136 ZA1RS1 MR XXXXXXXX XXXXXXXX XXXXXXXX
  MDISK F300 3390 8450 017 ZA1RS1 MW XXXXXXXX XXXXXXXX XXXXXXXX
  MDISK 0301 3390 8467 007 ZA1RS1 MR XXXXXXXX XXXXXXXX XXXXXXXX
  MDISK 0302 3390 8474 007 ZA1RS1 MR XXXXXXXX XXXXXXXX XXXXXXXX

*DVHOPT LNK0 LOG1 RCM1 SMS0 NPW1 LNGAMENG PWC20160516 CRC"y
DVHRXV3366I Preceding records were included from RACFVM-1 configuration
DVHRXV3366I for node ITS0ZVM1.
----- 53 line(s) not displayed -----
ACCOUNT SYSTEMS
IPL 490 PARM AUTO CR
IUCV *RPI PRIORITY MSGLIMIT 100
IUCV ANY PRIORITY MSGLIMIT 50
IUCV ALLOW MSGLIMIT 255
MACH XA
OPTION QUICKDSP MAXCONN 300
CONSOLE 0009 3215 T OPERATOR
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A

  MDISK 0200 3390 DEVNO 3B07 MWV XXXXXXXX XXXXXXXX XXXXXXXX
  MDISK 0300 3390 DEVNO 3C07 MWV XXXXXXXX XXXXXXXX XXXXXXXX
*DVHOPT LNK0 LOG1 RCM1 SMS0 NPW1 LNGAMENG PWC20160516 CRCü
DVHREV3356I The following are your user option settings:
DVHREV3356I Links DISABLED Logging ON RcvMsg ON Smsg OFF NeedPW ON
DVHREV3356I Lang AMENG
DVHREV3357I The following links are in effect to your virtual machine:
DVHREV3357I To your 0301 as their 0301, Mode MR by user ID RACMNT-1
DVHREV3357I To your 0302 as their 0302, Mode MR by user ID RACMNT-1
----- 6 line(s) not displayed -----
DVHREV3357I To your 0305 as their 0305, Mode RR by user ID IBMUSER
```

```
DVHREV3357I To your 0305 as their 0305, Mode RR by user ID SYSADMIN
DVHREV3357I To your 0200 as their 0200, Mode MW by user ID RACMNT-1
DVHREV3357I To your 0300 as their 0300, Mode MW by user ID RACMNT-1
----- 6 line(s) not displayed -----
* * * End of File * * *
```

Defining the RACF database disks as shared

The RACF database DASD must be defined to CP as shared by adding **RDEVICE** statements to the SYSTEM CONFIG file for the SSI cluster. SYSTEM CONFIG is on PMAINT CFO. For this example, we add the following statements to SYSTEM CONFIG:

```
RDevice 3B07 Type DASD Shared YES
RDevice 3C07 Type DASD Shared YES
```

Note: Run the **CPSYNTAX** utility over your SYSTEM CONFIG file after making any changes. In an SSI configuration, you must use the **LPAR** option to test the SSI multi-configuration nature of SYSTEM CONFIG. Run this once for each logical partition (LPAR) in your SSI configuration, even if you believe that you made a change that needs to be tested once.

Defining the initial RACF database

After defining the new full pack minidisks for the RACF database, you must initialize the disks. An easy way to do this is to copy the existing supplied database minidisks by using DDR. In this example, we did this by using the process that is shown in Example 4-5.

Example 4-5 Use DDR to initialize the RACF database disks

```
link racfvm f200 1200 rr
Ready; T=0.01/0.01 11:50:09
link racfvm 200 2200 w
Ready; T=0.01/0.01 11:52:59
ddr
z/VM DASD DUMP/RESTORE PROGRAM
ENTER:
sysprint cons
ENTER:
input 1200 dasd
ENTER:
output 2200 dasd scratch
ENTER:
copy all
HCPDDR711D VOLID READ IS RACF
DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:
yes
COPYING RACF
COPYING DATA 06/14/16 AT 15.54.06 GMT FROM RACF TO SCRATCH
INPUT CYLINDER EXTENTS OUTPUT CYLINDER EXTENTS
START STOP START STOP
0 16 0 16
END OF COPY
ENTER:

END OF JOB
Ready; T=0.01/0.01 11:54:12
det 1200 2200
1200 2200 DETACHED
```

```

Ready; T=0.01/0.01 11:55:36
link racfvm f300 1300 rr
Ready; T=0.01/0.01 11:55:52
link racfvm 300 2300 w
Ready; T=0.01/0.01 11:56:03
ddr
z/VM DASD DUMP/RESTORE PROGRAM
ENTER:
sysprint cons
ENTER:
input 1300 dasd
ENTER:
output 2300 dasd scratch
ENTER:
copy all
HCPDDR711D VOLID READ IS RACFBK
DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:
yes
COPYING RACFBK
COPYING DATA 06/14/16 AT 15.56.53 GMT FROM RACFBK TO SCRATCH
INPUT CYLINDER EXTENTS OUTPUT CYLINDER EXTENTS
      START      STOP      START      STOP
        0        16         0         16
END OF COPY
ENTER:

END OF JOB
Ready; T=0.01/0.01 11:56:56
det 1300 2300
1300 2300 DETACHED
Ready; T=0.01/0.01 11:57:01

```

The volume label that is read from the source disk in each DDR step helps you ensure that the correct disk is copied. “RACF” and “RACFBK” are the labels that are expected on the primary and backup disks.

Note: For more information about how to move from minidisk to full-pack if an increase in the database allocation is needed, see *RACF Security Server System Programmer’s Guide*, SC24-6312.

Defining RACF user IDs

The following VMs are defined in a default CP directory:

7VMRAC10	Product owning VM (This is a release-specific user ID and changes with every new release of z/VM.)
RACFVM	Production VM
RACFSMF	SMF VM
RACMAINT	Backup VM
IBMUSER	Initial RACF administrator
AUTOLOG1	System startup machine
AUTOLOG2	System startup machine

Users with NOLOG password

In a normal z/VM installation, several user IDs are defined with the password NOLOG. *RACF Security Server Security Administrator Guide*, SC24-6311 describes the following methods that can be used to handle such users:

- Leave the password as NOLOG.

RPIDIRCT defines these users as protected and revoked by setting the NOPASSWORD, NOPHRASE, and REVOKED attributes. However, because NOLOG is a special reserved password to CP that prevents access, a logon request from such a user cannot be passed to RACF. Therefore, if such a user must access the system in the future, the CP directory and RACF must be updated to activate the user.

- Change the password to UNLOG.

Similar to the NOLOG password, **RPIDIRCT** defines these users as protected and revoked by setting the NOPASSWORD, NOPHRASE, and REVOKED attributes. Because UNLOG is not a CP reserved password, CP passes a logon request for such a user to RACF. Therefore, the user can be granted system access in the future only by performing a RACF update.

What you decide to do here depends on the local security policy and procedure, and whether you intend to use the DirMaint-RACF connector. The password management code in the connector handles what to do with privileged passwords, such as NOLOG; therefore, changing the directory before setting up RACF yields little advantage.

Note: The use of the **RPIDIRCT** command creates a user with a NOLOG or UNLOG password with the NOPASS, NOPHRASE, and REVOKED set. However, the DirMaint-RACF connector sets only REVOKED for a NOLOG user. If your installation performs all password management by using DirMaint, this issue is not important because the connector resets the password field in RACF if the user is migrated out of NOLOG status.

If you decide to change the NOLOG passwords to UNLOG, proceed with “Changing NOLOG to UNLOG”. If you decide not to, continue with “Evaluating the minidisk access” on page 68.

Changing NOLOG to UNLOG

To perform this change, you must determine what file is used for the source directory while implementing RACF. If you implemented DirMaint, you must obtain a copy of the USER WITHPASS file from the DIRMAINT VM, as shown in Example 4-6.

*Example 4-6 DIRM USER WITHPASS***dirm user withpass**

```
DVHXMT1191I Your USER request has been sent for processing.
Ready; T=0.03/0.03 11:38:50
DVHREQ2288I Your USER request for MAINT at * has been accepted.
RDR FILE 0012 SENT FROM DIRMAINT PUN WAS 0020 RECS 2261 CPY 001 A NOHOLD
DVHREQ2289I Your USER request for MAINT at * has completed; with RC = 0.
```

receive 12 user withpass a2 (replace

```
File USER WITHPASS A2 replaced USER WITHPASS A0 with USER WITHPASS A0 rec
from DIRMAINT at VMLINUX5
Ready; T=0.01/0.01 11:39:15
```

If you did not implement DirMaint, you can use a copy of the USER DIRECT file on MAINT's 2CC disk. When you receive this file, save it as an A2 file to allow the RACF user ID to access the file in later steps. This file is required when the **RPIDIRCT EXEC** runs later.

The easy way is to perform a global XEDIT **change** command and change NOLOG to UNLOG within the file, as shown in Figure 4-2.

```
USER      WITHPASS A2  F 80  Trunc=80 Size=220
====> ch /NOLOG/UNLOG/*
    90 USER $ALLOC$  NOLOG
    96 USER $DIRECT$ NOLOG
   100 USER $SYSCKP$ NOLOG
   104 USER $SYSWRM$ NOLOG
   108 USER $PAGE$   NOLOG
   112 USER $SPOOL$  NOLOG
   116 USER $TDISK$  NOLOG
   728 USER ROOT NOLOG 32M 32M G
   732 USER DAEMON NOLOG 32M 32M G
   736 USER BIN NOLOG 32M 32M G
   740 USER SYS NOLOG 32M 32M G
   744 USER ADM NOLOG 32M 32M G
   748 USER NOBODY NOLOG 32M 32M G
   752 USER DEFAULT NOLOG 32M 32M G
  2203 * * * End of File * * *
```

Figure 4-2 USER WITHPASS

Evaluating the minidisk access

A z/VM system includes several user minidisks with a read password of ALL, which means that the disk is accessible to all users on the system without prompting for a password. This is usually for disks that contain programs that can be used by all users on the system. MAINT 190 and TCPMAINT 592 are examples (CMS and TCP/IP clients).

The **RPIDIRCT EXEC** command, which is used in the RACF installation process to create RACF authorization commands from CP directory entries, uses the UACC (universal access) attribute of the created resource to make an equivalent, as shown in the following example:

```
RDEFINE VMMDISK MAINT.190 OWNER(MAINT) UACC(READ)
```

Auditors often flag any profile with UACC(READ) as a potential area for information leakage, and recommend UACC(NONE) be used instead. It is a preferred practice to use a PERMIT ACL specifying ID(*) in place of UACC(READ). Review the passwords on your system before running **RPIDIRCT**.

Running RPIDIRCT

RPIDIRCT EXEC is used to generate the RPIDIRCT SYSUT1 file that contains all the RACF commands to add the users to define the RACF classes, such as VMMDISK and VMRDR, and to permit the owners to the resources. This exec is run from the product owning VM for RACF.

Before you run the exec, you must obtain a copy of the current user directory. If you do not use a directory manager (such as DirMaint), the directory is contained in the file `USER DIRECT` on `PMAINT 2CC`.

If you use DirMaint, you must run the command **DIRM USER WITHPASS** from a DirMaint administration user ID, and make the resulting `USER WITHPASS` file available to the `7VMRAC10` user. How we performed this on our system is shown in Example 4-7.

Example 4-7 Run DIRM USER WITHPASS

```
dirm user withpass
DVHXMT1191I Your USER request has been sent for processing to DIRMAINT
DVHXMT1191I at ITS0ZVM1.
Ready; T=0.01/0.01 15:39:48
  DVHREQ2288I Your USER request for VIC at * has been accepted.
RDR FILE 0065 SENT FROM DIRMAINT PUN WAS 4381 RECS 5673 CPY 001 A NOHOLD NOKEEP
  DVHREQ2289I Your USER request for VIC at * has completed; with RC = 0.
receive 65
File USER WITHPASS A0 created from USER WITHPASS A0 received from DIRMAINT at IT
SOZVM1
Ready; T=0.01/0.01 15:39:54
sendf user withpass a to 7vmrac10
File USER WITHPASS A0 sent to 7VMRAC10 at ITS0ZVM1 on 06/12/19 15:40:01
Ready; T=0.01/0.01 15:40:02
```

On `7VMRAC10`, we used the **RECEIVE** command to accept the file that is sent from our administrator user. You also must access the `7VMRAC10 651` disk, which is where the **RPIDIRECT EXEC** is stored.

Because **RPIDIRECT EXEC** generates much output, run the **cp term more 0 0** command before running the exec. This process makes the exec run without you needing to clear the pane repeatedly. However, if you run this command, spool your terminal in case an error occurs and you must discover what happened. Preparation for running **RPIDIRECT** is shown in Figure 4-3.

```
RDR FILE 0005 SENT FROM VIC          PUN WAS 0069 RECS 5673 CPY 001 A
NOHOLD NOKEEP

receive 5
File USER WITHPASS A0 created from USER WITHPASS A0 received from
VIC at ITS0ZVM1
Ready; T=0.01/0.01 15:46:22

acc 651 e
Ready; T=0.01/0.01 15:48:58

cp spool console * start
Ready; T=0.01/0.01 15:49:14

cp term more 0 0
```

Figure 4-3 Set up to run RPIDIRECT EXEC

When you run the **RPIDIRECT EXEC** command, you must provide the file name and file type of the source directory file. It searches for the file on all accessed disks.

The default output file mode is A. When the exec starts, it prompts you to change the default group ID. Because we used the default, we replied N to the prompt question.

Figure 4-4 shows an example of the **rpidirct user withpass** command.

```

USER WITHPASS Filemode defaulted to "*".
Output defaulted to "A" disk.
  Default group ID = SYS1.
  Would you like to change this default?
  Enter Y/N
N
  Default group ID = SYS1.

PROFILE IBMDFLT

PROFILE TCPCMSU

PROFILE TCPGCSU
----- 4277 line(s) not displayed -----

***** 5531 Directory records processed *****

***** RPIDIRCT SYSUT1 CREATED *****

```

Figure 4-4 Run **RPIDIRCT EXEC**

Secure Configuration Guide for z/VM, SC24-6323 suggests that, after running **RPIDIRCT**, you modify the resulting **RPIDIRCT SYSUT1** file in the following ways:

- ▶ Alter the VMRDR profile for MAINT710 to specify UACC (UPDATE).
- ▶ Add any PERMITs that are required.

Updating the MAINT710 profile in VMRDR is required for the correct operation of the **SERVICE EXEC**, as described in *Program Directory for RACF Security Server for z/VM*, GI13-4358.

Make the same changes to the following VMs:

TCPMAINT	The TCP/IP daemon VMs spool their console to TCPMAINT.
DIRMAINT	Makes the DIRM SEND command work.
DATAMOVE	Allows DIRMAINT to send files to DATAMOVE (2 3 4 too).
DIRMSATn	Allows the correct DirMaint operation across the SSI cluster.

Other permits are required.

In our installation, several guests used the **COMMAND** directory control statement to include a **SET VSWITCH GRANT** command that authorized access to a VSWITCH. These control statements can be seen in the USER WITHPASS file, as shown in the following example:

```
COMMAND SET VSWITCH VSW1 GRANT &USERID
```

To make sure that these statements are considered, complete the following steps:

1. Find all **SET VSWITCH GRANT** commands in the CP directory.
2. Scan the **RPIDIRCT SYSUT1** file to see whether **RDEFINE** commands for the VSWITCH or Guest LAN that is named on the **SET VSWITCH GRANT** commands are present (if a **NICDEF** directory control statement mentions the same VSWITCH, **RPIDIRCT** created the **RDEFINE** command).

If none exists, add the appropriate **RDEFINE** commands to RPIDIRCT SYSUT1, as shown in the following example:

```
RDEFINE VMLAN SYSTEM.VSW1 UACC(NONE)
```

3. For each **SET VSWITCH GRANT** command:

- a. If it appears in a directory PROFILE entry, make a list of all users that **INCLUDE** that profile entry.
- b. Add the appropriate **PERMIT** command (or commands) to RPIDIRCT SYSUT1. If the command is on a single directory entry, a single **PERMIT** command is needed. If the command is part of a directory profile, add a **PERMIT** command for each user that has an **INCLUDE** statement for that profile.

```
PERMIT SYSTEM.VSW1 CLASS(VMLAN) ID(LNXS0006) ACC(UPDATE)
```

In addition, if the **SET VSWITCH GRANT** command includes the VLAN parameter, complete the following steps:

- i. Scan the RPIDIRCT SYSUT1 file to see whether an **RDEFINE** command exists for the VLAN ID on the VSWITCH or Guest LAN. If none exists, add the appropriate **RDEFINE** commands to RPIDIRCT SYSUT1 (a separate **RDEFINE** is needed for each VLAN that is on the **SET VSWITCH GRANT** command), as shown in the following example:

```
RDEFINE VMLAN SYSTEM.VSW1.0201 UACC(NONE)
```

- ii. Add the appropriate **PERMIT** command (or commands) to RPIDIRCT SYSUT1, as shown in the following example:

```
PERMIT SYSTEM.VSW1.0201 CLASS(VMLAN) ID(LNXS0006) ACC(UPDATE)
```

A similar process is used for virtual network access that is granted by using MODIFY statements in SYSTEM CONFIG or **SET VSWITCH GRANT** commands in AUTOLOG1 PROFILE EXEC or other scripts.

For each statement or command, complete the following steps:

1. Scan RPIDIRCT SYSUT1 to see whether an **RDEFINE** for the VSWITCH or Guest LAN exists. If none is present, add the appropriate **RDEFINE** command to RPIDIRCT SYSUT1, as shown in the following example:

```
RDEFINE VMLAN SYSTEM.VSW1 UACC(NONE)
```

2. Add the appropriate **PERMIT** command to RPIDIRCT SYSUT1, as shown in the following example:

```
PERMIT SYSTEM.VSW1 CLASS(VMLAN) ID(LNXS0006) ACC(UPDATE)
```

In addition, if the **SET VSWITCH GRANT** command includes the VLAN parameter, complete the following steps:

- a. Scan the RPIDIRCT SYSUT1 file to see whether an **RDEFINE** command exists for the VLAN ID on the VSWITCH or Guest LAN. If none is present, add the appropriate **RDEFINE** commands to RPIDIRCT SYSUT1 (a separate **RDEFINE** is needed for each VLAN given on the **SET VSWITCH GRANT** command), as shown in the following example:

```
RDEFINE VMLAN SYSTEM.VSW1.0201 UACC(NONE)
```

- b. Add the appropriate **PERMIT** command (or commands) to RPIDIRCT SYSUT1, as shown in the following example:

```
PERMIT SYSTEM.VSW1.0201 CLASS(VMLAN) ID(LNXS0006) ACC(UPDATE)
```

Note: If many guest virtual network connections or a complex virtual network configuration exist, you might decide to leave activating the VMLAN class until after the rest of RACF configuration is stabilized.

You can defer the activation of VMLAN until a later time by commenting out the **SETOPTS CLASSACT(VMLAN)** command from the RPIDIRECT SYSUT1 file. Resource profiles and permit statements that are contained in RPIDIRECT SYSUT1 are defined to the RACF database, but the class is not activated; therefore, your virtual network permission structure is retained.

Take the time to implement VMLAN when the RACF is installed. It might not be feasible to return at a later and implement the required changes.

Customizing the processing of SMF records

One of the reasons that you run RACF on your z/VM system is to audit who is doing what on the system. This auditing requires the configuration of SMF to record reliably the audit information that is captured by RACF. To use the RACFSMF VM, you must set up the PROFILE EXEC and the SMF CONTROL files by completing the following tasks:

- ▶ Create the RACFSMF PROFILE EXEC.
- ▶ Modify the SMF CONTROL file.

These tasks are described next.

Creating the RACFSMF PROFILE EXEC

You create the PROFILE EXEC by copying the SMFPROF EXEC from the RACFVM 305 minidisk, as shown in Example 4-8.

Example 4-8 Create the PROFILE EXEC for RACFSMF

```
link racfvm 305 305 rr
DASD 0305 LINKED R/O; R/W BY RACFVM   at ITS0ZVM4
Ready;
  acc 305 e
DMSACP723I E (305) R/O
Ready;
  link racfsmf 191 291 mr
Ready;
  acc 291 m
Ready;
  copy smfprof exec e profile exec m
Ready;
```

After you copy the file, modify the **SMFFREQ** and **SMFSWTCH** parameters to match what is shown in Example 4-9.

Example 4-9 RACFSMF's PROFILE EXEC

```
PROFILE EXEC      M1 V 130 Trunc=130 Size=428 Line=124 Col=1
====>
124 Smfpct      = 80
125 Smfinfo     = 'OPERATOR'      /* Default message \r
126 Smffreq     = ' AUTO '        /* Valid values: DAILY, WEEKLY,
127                                     /*                                     AUTO
128 Smfday       = 'MONDAY'        /* Valid values: SATURDAY - FRI
129 Smfswtch    = ' NO '         /* Valid values: YES NO
130 /* 1 line deleted
```

```
131 hi = '1de8'x
132 lo = '1d60'x
```

Modifying the SMF CONTROL file

The Program Directory instructs you to detach the RACFSMF 191 disk when you complete the work on the PROFILE EXEC. However, the next step instructs you to link and access this disk again because you must copy the SMF CONTROL file to several disks.

The SMF CONTROL file is on the RACFVM 191 disk. The directions instruct you to copy it to the RACFSMF 191 disk, modify it, and then copy it back to the original disk. It is easier to modify the one on the RACFVM disk and then copy it to the two other disks.

Link and access the appropriate disk. Because you still have the RACFSMF 191 disk, you can complete the steps that are shown in Example 4-10.

Example 4-10 Access the appropriate disk

```
link racfvm 191 391 mw
DASD 0391 LINKED R/W; R/W BY RACFVM   at ITS0ZVM4
Ready;
  link racmaint 191 491 mr
Ready;
  acc 391 n
Ready;
  acc 491 o
Ready;
```

Edit the SMF CONTROL file on the N disk (which is the RACFVM 191 disk). Make the change that is shown in Example 4-11. The **SEVER** keyword determines RACF behavior if the SMF disks are filled. If **SEVER** is set to NO, auditing continues with newer SMF data overwriting older audit records. If **SEVER** is set to YES, RACF ceases operations because it cannot audit security relevant events on the hypervisor. The **SEVER** keyword is initially set to NO. If you choose to set **SEVER** to YES, RACF severs the path between CP and RACF when the SMF disks are full, and RACF cannot continue recording SMF records.

The file contains only one line, so it is split into two lines for readability.

Example 4-11 SMF CONTROL

```
SMF      CONTROL N1 F 100 Trunc=100 Size=2
====>
* * * Top of File * * *
CURRENT 301 K PRIMARY 301 K SECONDARY 302 K
10000 VMSP CLOSE 001 SEVER YES 0 RACFSMF
* * * End of File * * *
```

After modifying this file, you must copy it to the *M* and *O* disks. Then, the **flist smf control** * command should return results similar to the results that are shown in Example 4-12.

Example 4-12 File list of SMF CONTROL

LVL	0	-----	SMF	CONTROL	*-----	FILE	1	OF	4
SMF		CONTROL	E1	F	100	1	1	11/29/05	12:57
SMF		CONTROL	M1	F	100	1	1	6/22/16	14:52
SMF		CONTROL	N1	F	100	1	1	6/22/16	14:40
SMF		CONTROL	O1	F	100	1	1	6/22/16	14:51

The SMF CONTROL file on the E disk is your original file on the RACFVM 305 disk, and it should not be changed. Now, you can detach the 291, 391, and 491 disks.

Password encryption algorithm

In z/VM 6.3, RACF for z/VM introduced a new password encryption method that is known as *KDFAES*. This method uses strong encryption and is supported by the hardware CPACF feature to protect the RACF database from an offline attack. As a preferred practice, use KDFAES. To activate KDFAES, run the following command:

```
SETRPTS PASSWORD(ALGORITHM(KDFAES))
```

Note: For more information about enabling and working with KDFAES, including considerations for enabling this mode, see “RACF KDFAES algorithm” on page 122.

If KDFAES is not used, the RACF exit ICHDEX01 controls whether masking or DES encryption is used for password encryption. If the IBM supplied ICHDEX01 exit is present and active, RACF password masking is used. If the ICHDEX01 exit is deactivated or not present, RACF DES encryption is used.

RACF DES encryption is recommended over the masking technique. Before RACF function level 540 (z/VM 5.4), the ICHDEX01 exit was deleted to allow RACF DES encryption to occur. From RACF FL540 onward, the ICHDEX01 exist is disabled by default.

Removing ICHRCX02

As a preferred practice, remove the ICHRCX02 exit that is enabled by default. ICHRCX02 removal disables batch-mode alternative user ID user support.

To perform this step, you do *not* need the HLASM. To delete the ICHRCX02 exit, follow the instructions in Appendix B.3, “Local Modification to Full Part Replacement Text Files”, in the *Program Directory for RACF Security Server for z/VM*, GI13-4358 by using the following substitution values:

- For fn, use ICHRCX02
- For blist, use RPIBLLPA

Use the VMSES/E process to create a local modification to this load library. A local copy of the **RPIBLLPA EXEC** is created and the local modification is logged in the local version vector table for the product. The local version vector table is a log file of the parts for which you performed local service. It is important to complete these steps so that future IBM service to this part does not overlay your local modifications.

The first step in deleting this member of the RACFLPA LOADLIB is to establish the 7VMRAC10 minidisk order. In this example, we used the VMSES/E exec VMFSETUP to perform this step (see Example 4-13).

Example 4-13 VMFSETUP for RACF

```
vmfsetup 7VMRAC10 racf
VMFSET2760I VMFSETUP processing started for 7VMRAC10 RACF
VMFUTL2205I Minidisk|Directory Assignments:
          String      Mode  Stat  Vdev  Label  (OwnerID Odev : Cyl/%Used)
          -or-          SFS Directory Name
VMFUTL2205I LOCALSAM  E     R/W   2C2   RAC2C2 (7VMRAC10 02C2 :   9/00)
VMFUTL2205I APPLY     F     R/W   2A6   RAC2A6 (7VMRAC10 02A6 :   9/01)
VMFUTL2205I           G     R/W   2A2   RAC2A2 (7VMRAC10 02A2 :   9/01)
VMFUTL2205I DELTA     H     R/W   2D2   RAC2D2 (7VMRAC10 02D2 :  70/01)
VMFUTL2205I BUILD0    I     R/W   29E   RAC29E (7VMRAC10 029E : 10/50)
```



```

VMFUTL2205I BUILD6 J R/W 599 RAC599 (7VMRAC10 0599 : 31/45)
VMFUTL2205I BUILD4 K R/W 505 RAC505 (7VMRAC10 0505 : 41/71)
VMFUTL2205I BUILD2 L R/W 590 RAC590 (7VMRAC10 0590 : 63/33)
VMFUTL2205I BUILD8 M R/W 651 RAC651 (7VMRAC10 0651 : 1/23)
VMFUTL2205I BASE N R/W 2B2 RAC2B2 (7VMRAC10 02B2 : 85/84)
VMFUTL2205I ----- A R/W 191 MNT191 (MAINT710 0191 : 175/01)
VMFUTL2205I ----- B R/W 5E6 MNT5E6 (MAINT710 05E6 : 9/79)
VMFUTL2205I ----- C R/W 2CC MNT2CC (PMAINT 02CC : 10/16)
VMFUTL2205I ----- D R/W 51D MNT51D (MAINT710 051D : 26/22)
VMFUTL2205I ----- S R/O 190 MNT190 (MAINT 0190 : 207/47)
VMFUTL2205I ----- Y/S R/O 19E MNT19E (MAINT 019E : 500/39)
VMFSET2760I VMFSETUP processing completed successfully
Ready; T=0.03/0.04 11:52:06

```

The next step is to determine the highest level of service to the build list for the RACFLPA load library by using the **VMFSIM EXEC** with the **GETLVL** parameter. The exec searches all of the version vector tables for this product and determines the highest level of service. It returns the file name and file type of that part. If you do not run the **VMFSETUP** exec before you run the **VMFSIM** exec, you do not receive the correct results.

Example 4-14 shows the **vmfsim getlvl** command. It gives you the file name and file type of the file that you need to copy to create your file.

Example 4-14 The vmfsim getlvl command

```

vmfsim getlvl 7VMRAC10 RACF tdata :part rpibllpa exc (history
:PART RPIBLLPA EXC00000 BASE-FILETYPE
Ready; T=0.01/0.01 11:54:54

```

The output from the **vmfsim getlvl** command lists this element as **BASE-FILETYPE**. In VMSES/E terminology, it means that no service was made to this part by IBM or locally by a system programmer (no entries in the IBM and Local Version Vector Tables). In our case, we use the **RPIBLLPA EXEC**. You must determine on which disk the base file is stored. Copy the highest level of the build list to the 2C2 local disk (E-disk).

Use the following syntax:

```
copyfile blist ft * = exclnnnn 2c2_fm
```

Where **blist** is the file name to be copied, **ft** is the file type, **nnnn** is a local tracking number that you supply, and **2c2_fm** is the file mode of the 2C2 minidisk. Because this modification is the first modification to this file, we use 0001 as the **nnnn** value and file mode **e** to reflect the 2C2 minidisk, as shown in the following example:

```
copyfile rpibllpa exec u = excl0001 e
```

Modify the **RPIBLLPA EXCL0001** on the E disk and comment out the entry for the **ICHRCX02** member, as shown in Example 4-15.

Example 4-15 RPIBLLPA EXCL0001

```

RPIBLLPA EXCL0001 E1 F 80 Trunc=80 Size=749 Line=456 C
====>
456 *
457 *:OBJNAME. ICHRCX02 LEPARMS RENT REUS LET NCAL XREF
458 *:BLDREQ. RPIBLOBJ. ICHRCX02
459 *:OPTIONS. CONCAT SYSLIB RACFOBJ
460 *:OPTIONS. INCLUDE RACFOBJ( ICHRCX02)

```

```

461 *:OPTIONS. ENTRY ICHRCX02
462 *:EOBJNAME.
463 *

```

Log this local modification to the **RPIBLLPA EXEC** into the local version vector table. In earlier releases of z/VM, the **VMFSIM MODIFY** command was used. Starting with z/VM 5.2.0, you can use the **VMFSIM LOGMOD** command with more user-friendly syntax, as shown in the following example:

```
vmfsim logmod 7VMRAC10 vvtlcl e tdata :mod lcl0001 :part rpibllpa exc
```

The 2C2 disk should now contain 7VMRAC10 VVTLCCL and RPIBLLPA EXCL0001 files. Example 4-16 shows the content of the 7VMRAC10 file.

Example 4-16 File list of the 2C2 disk

```

7VMRAC10 FILELIST A0  V 169  Trunc=169 Size=2 Line=1 Col=1 Alt=0
Cmd  Filename Filetype Fm Format Lrec1      Records      Blocks      Date
     7VMRAC10 VVTLCCL  E1 V          32          1          1 6/14/16
     RPIBLLPA EXCL0001 E1 F          80         749         15 6/14/16

7VMRAC10 VVTLCCL  E1  V 80  Trunc=80 Size=1
====>
  0 * * * Top of File * * *
  1 :PART.RPIBLLPA EXC :MOD.LCL0001
  2 * * * End of File * * *

```

Next, generate a new RACFLPA LOADLIB by using the **VMFBLD** command. When you run the command, make sure that you specify the **blist** parameter (in this case, **rpibllpa**), as shown in the following example:

```
vmfbld ppf 7VMRAC10 RACF rpibllpa (all)
```

If you do not specify this parameter, all build lists that are listed in the BLD section of the 7VMRAC10 PPF file are built (see Example 4-17).

Example 4-17 VMFBLD process for loadlib

```

VMFBLD2195I VMFBLD PPF 7VMRAC10 RACF RPIBLLPA ( LOG CNTRL RPIVM NOCKVV
          NOSETUP ALL
VMFBLD2760I VMFBLD processing started
VMFUTL2205I Minidisk|Directory Assignments:
          String      Mode Stat Vdev Label/Directory
VMFUTL2205I LOCALSAM  E    R/W 2C2  RAC2C2
----- 13 line(s) not displayed -----
VMFBLD1851I Reading build lists
VMFBLD2182I Identifying new build requirements
VMFBLD2182I New build requirements identified
VMFBLD1851I (1 of 1) VMFBDLLB processing RPIBLLPA EXCL0001 E, target
          is BUILD4 505 (K)
VMFLLB2217I RACFLPA LOADLIB will be rebuilt because all members must
          be rebuilt
----- 66 line(s) not displayed -----
VMFBLD1851I (1 of 1) VMFBDLLB completed with return code 0
VMFBLD2180I There are 0 build requirements remaining
VMFBLD2760I VMFBLD processing completed successfully

```

To place the new local modification into production, you must link to the RACFVM 305 disk and then use the **VMFCOPY** command to copy the files to the production disk (see Example 4-18). The **VMFCOPY** updates the VMSES PARTCAT file on the 305 disk.

Example 4-18 Place changes into production

```
link RACFVM 305 305 MR
acc 505 e
acc 305 f
vmfcopy RACFLPA * e = f (prodid 7VMRAC10%RACF replace oldd)
```

4.2.2 Building the RACF enabled CPLOAD MODULE

Make sure that you logged off from the RACF product owner VM and logged on to the MAINT710 VM. When the **PROFILE EXEC** completes running, you have all the required disks that are accessed.

The RACF product is included with the system in a disabled state. You can use the VMSES/E command **SERVICE** to enable the product. You also can generate a CPLOAD MODULE that enables RACF to CP. This new CP nucleus requires that RACF is active on the system to manage authentication. The

initial settings for RACF are that if a resource is not defined to RACF, the decision on the access request is deferred to CP. Later in this setup process, you change this setting to secure the system so that all resources must be defined to RACF or the request for access fails.

Run the **service racf enable** command. Example 4-19 shows the output.

Note: The new CP nucleus, with the RACF CP parts, is placed on the secondary parm disk (MAINT710 CF2). For your information, a copy of the previous (or currently running) CPLOAD MODULE is still on the primary (CF1) and tertiary (CF3) parm disks as CPLOAD MODULE. It is also saved on the secondary parm disk as CPLOLD MODULE.

Example 4-19 CP SET PRODUCT

```
VMFSRV2195I SERVICE RACF ENABLE
VMFSRV2760I SERVICE processing started
VMFSRV2767I Reading VMFINS DEFAULTS B for additional options
VMFSRV2760I VMFINS processing started
VMFSRV2602R The following components can be enabled for PROD 7VMRAC10
RACF. Enter the number of your choice
(0) Bypass this product
(1) :PPF 7VMRAC10 RACF :PRODID 7VMRAC10%RACF
:DESC RACF Feature of z/VM, FL710
(2) :PPF SERVP2P RACF :PRODID 7VMRAC10%RACF
:DESC RACF Feature of z/VM, FL710
(3) Exit
VMFINS2603I Processing product :PPF 7VMRAC10 RACF :PRODID
7VMRAC10%RACF
VMFINS2603I Enabling product 7VMRAC10
VMFINS2771I The CP SET PRODUCT command completed successfully for product 7VMRAC10
VMFINS2772I File 7VMRAC10 PRODSYS created on your A-disk contains the
system configuration PRODUCT statement for product 7VMRAC10
VMFINS2603I PRODUCT ENABLED IN VMSE/E, CP PROCESSING REQUIRED
```

```
VMFINS2760I VMFINS PROCESSING COMPLETED SUCESSFULLY
HCPZAC6730I CPRELEASE REQUEST FOR DISK A COMPLETED.
```

When the product is enabled dynamically, the configuring of RACF by the service exec sets a flag in a VMSES/E software inventory table. This flag causes the CP nucleus to be built by using the RACF versions of the HCPRWA, HCPRPD, HCPRPW, HCPRPI, HCPRPG, and HCPRPF files. The **SERVICE EXEC** then generates a new CPLOAD MODULE and places it on the CF2 disk only (it is moved to the other parm disks in a later step).

When the **SERVICE EXEC** completes, issue the **VMFVIEW** command to verify that no problems exist.

Run an IPL of your system with RACF in test mode

To prepare for the next step, you also must find the device address of the volume on which the alternative parm disk is (MAINT710 CF2; on our system, it was on the 710RL1 volume). Shut down your system and then, by using the **LOADPARM** option, run an IPL of your system again. The z/VM stand-alone program loader starts.

Note: When z/VM SSI was introduced, the locations and roles of the parm disks changed. Previously, all of the parm disks were owned by MAINT and contained the CP nucleus, system configuration file, and logo configuration. With SSI, a new parm disk that is owned by the PMAINT user holds the system and logo configuration files, a second parm disk that is owned by the MAINT710 user is used during the service process, and a separate pair of disks that is owned by MAINT on each member of the SSI cluster holds the CP nucleus for that member.

You must start z/VM by using the new RACF-enabled CP nucleus that was generated by the SERVICE process. From the SAPL pane, enter the device address of the alternative parm disk volume, as shown in Example 4-20.

Example 4-20 IPL from CPLOAD on alternative parm disk

```
STAND ALONE PROGRAM LOADER: z/VM VERSION 7 RELEASE 1.0
```

```
DEVICE NUMBER:  03234      MINIDISK OFFSET:  209      EXTENT:  1
```

```
MODULE NAME:    ZVM710R    LOAD ORIGIN:      1000
```

```
-----IPL PARAMETERS-----
```

```
fn=SYSTEM ft=CONFIG pdnum=1 pdvo1=3031
```

```
-----COMMENTS-----
```

```
-----
9= FILELIST  10= LOAD  11= TOGGLE EXTENT/OFFSET
```

Note: If Auto_Warm_IPL is coded in your SYSTEM CONFIG file, you must also add the IPL parameter **PROMPT** on the SALIPL pane.

You can verify that you can access the correct module by pressing PF9 to show the file list of the designated parm disk. The file list should look something the output that is shown in Example 4-21.

Example 4-21 File list of the alternative parm disk

STAND ALONE PROGRAM LOADER: z/VM VERSION 7 RELEASE 1.0								
FILENAME	FILETYPE	FORMAT	LRECL	RECORDS	BLOCKS	DATE	TIME	
CPLOAD	MODULE	Z1 V	65535	189	2999	6/14/19	17:07:42	
CPLD	MODULE	Z1 V	65535	189	2993	6/12/19	19:34:45	
3=QUIT 4=SORT(TYPE) 5=SORT(DATE) 6=SORT(NAME) 7=BACK 8=FORWARD 11=SELECT								

The CPLOAD MODULE includes a recent time stamp and is slightly larger than the CPLD MODULE (which is the previous non-RACF CP nucleus). Press PF3 to return to the SAPL pane.

When everything is ready, press PF10 to start the IPL.

During the IPL process, you must perform a **NOAUTOLOG** start and change the time of day if required (see Example 4-22). The **NOAUTOLOG** option tells the system *not* to start the AUTOLOG1 VM. Therefore, no other VMs are started automatically.

Example 4-22 z/VM IPL

```

09:07:02 z/VM V7 R1.0 SERVICE LEVEL 1801 (64-BIT)
09:07:02 SYSTEM NUCLEUS CREATED ON 2019-01-23 AT 12:25:50, LOADED FROM RD1RES
09:07:02
09:07:02 *****
09:07:02 * LICENSED MATERIALS - PROPERTY OF IBM* *
09:07:02 * *
09:07:02 * 5741-A09 (C) COPYRIGHT IBM CORP. 1983, 2018. ALL RIGHTS *
09:07:02 * RESERVED. US GOVERNMENT USERS RESTRICTED RIGHTS - USE, *
09:07:02 * DUPLICATION OR DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE *
09:07:02 * CONTRACT WITH IBM CORP. *
09:07:02 * *
09:07:02 * * TRADEMARK OF INTERNATIONAL BUSINESS MACHINES. *
09:07:02 *****
09:07:02
09:07:02 HCPZC06718I Using parm disk 1 on volume RD1RES (device 0123).
09:07:02 HCPZC06718I Parm disk resides on cylinders 209 through 248.
09:07:02 Start ((Warm|Force|COLD|CLEAN) (DRain) (DIsable) (NODIRect)
09:07:02 (NOAUTOlog)) or (SHUTDOWN)
09:07:08 NOAUTOLOG
09:07:08 NOW 09:07:08 EDT TUESDAY 2019-06-18
09:07:08 Change TOD clock (YES/NO)
NO

```

When the IPL completes, you start RACMAINT VM with the **xautolog racmaint** command. The reason for starting RACMAINT instead of RACFVM is that in a later step you run the **PUT2PROD** exec. This exec copies files to the RACFVM VM disks. RACMAINT links to those disks to run in READ ONLY mode, which allows MAINT and the **PUT2PROD** exec to gain write access to the disks owned by RACFVM.

When the RACMAINT VM logs on and runs the **PROFILE EXEC**, it runs **RACSTART EXEC**. This process causes this VM to be defined as the ESM for your system. You can ignore the messages about the 591 and 505 disk not being accessed. This issue does not cause a problem. You can now disconnect from the OPERATOR VM.

4.2.3 Updating the RACF database and options

The following tasks must be completed to update the RACF database with information from the CP directory and to set up options for the RACF environment.

Updating the RACF database with an existing CP directory

Log on to the IBMUSER VM. This VM is defined to have RACF special and operations authority in the initial RACF database that was included with the system. The password for this VM is SYS1, and you must change the password the first time that you log on.

From the VM, complete the following tasks:

1. Set a PF key to retrieve commands.
2. Run **RPIBLDDS** to build the RACF database.
3. Define the security administrator.

Before you can build the RACF database, you must link to the following the product owners' disks and access them (see Example 4-23):

- ▶ 191: Location of the RPIDIRECT SYSUT1 file
- ▶ 305: Location of the **RPIBLDDS EXEC**
- ▶ 29E: Location of the **RAC EXEC**

Example 4-23 Setting up the IBMUSER virtual machine

```
set pf12 retrieve
Ready; T=0.01/0.01 11:25:46
link 7vmrac10 505 305 rr
RPIMGR031E Resource 7VMRAC10.505 SPECIFIED BY LINK COMMAND NOT FOUND
DASD 305 LINKED R/O; R/W BY RACMAINT
Ready; T=0.01/0.01 11:33:40
link 7vmrac10 191 192 rr
RPIMGR031E Resource 7VMRAC10.191 SPECIFIED BY LINK COMMAND NOT FOUND
Ready; T=0.01/0.01 11:33:59
link 7vmrac10 29e 29e rr
RPIMGR031E Resource 7VMRAC10.29E SPECIFIED BY LINK COMMAND NOT FOUND
Ready; T=0.01/0.01 11:34:06
ac 305 c
ac 192 b
ac 29e d
```

Note: On our system, we found that IBMUSER included a disk that is linked at 305, which we detached so that we can perform these steps.

The **RPIBLDDS EXEC** is used to modify the RACF DATABASE. It uses the RPIDIRECT SYSUT1 file as input. This file was created earlier by the 7VMRAC10 VM with the **RPIDIRECT EXEC**. It contains all the RACF commands to add users, define resources, and authorize users to resources.

Example 4-24 shows RPIBLDDS being run.

Example 4-24 Run RPIBLDDS

rpibldds

Using default file RPIDIRECT SYSUT1

Processing batch file RPIDIRECT SYSUT1 using "RAC" command interface

=> RDEFINE VMCMD RACF UACC(READ)

=> RDEFINE VMCMD RAC UACC(READ)

=> ADDGROUP SYSTEM

=> ADDGROUP SYSTEM OVM(GID(0))

=> ADDGROUP STAFF

=> ADDGROUP STAFF OVM(GID(1))

=> ADDGROUP GBIN

. . .

=> RDEFINE VMMDISK MAINT710.5A2 OWNER(MAINT) UACC(NONE)

=> PERMIT MAINT710.5A2 CLASS(VMMDISK) RESET ID(MAINT710) AC(ALTER)

=> RDEFINE VMMDISK MAINT710.5A4 OWNER(MAINT) UACC(NONE)

=> PERMIT MAINT710.5A4 CLASS(VMMDISK) RESET ID(MAINT710) AC(ALTER)

=> RDEFINE VMMDISK MAINT710.5A6 OWNER(MAINT) UACC(NONE)

=> PERMIT MAINT710.5A6 CLASS(VMMDISK) RESET ID(MAINT710) AC(ALTER)

. . .

=> RDEFINE VMMDISK MAINT710.400 OWNER(LOHCOST) UACC(NONE)

=> PERMIT MAINT710.400 CLASS(VMMDISK) RESET ID(LOHCOST) AC(READ)

. . .

When the **RPIBLDDS EXEC** completes, your RACF database is initialized with all the VMs and resources that were included with the z/VM system. Now, you create more RACF administrators. You must determine what VMs are trusted to manage your secure environment.

The user IDs that are part of the system maintenance process (VMSES/E) must have authority to access minidisks across the system. For this reason, the RACF Program Directory recommends the following VMs be granted OPERATIONS authority:

- ▶ MAINT710
- ▶ BLDSEG
- ▶ BLDRACF
- ▶ BLDNUC
- ▶ BLDCMS
- ▶ MIGMAINT

The RACF **altuser** command is used to modify the RACF attributes for a VM (see Example 4-25).

Example 4-25 Set RACF attributes

rac alu MAINT710 operations

Ready; T=0.01/0.01 11:49:44

rac alu bldseg operations

Ready; T=0.01/0.01 11:49:53

rac alu bldracf operations

Ready; T=0.01/0.01 11:50:02

rac alu bldnuc operations

Ready; T=0.01/0.01 11:50:04

```
rac alu bldcms operations
Ready; T=0.01/0.01 11:50:07
rac alu migmaint operations
Ready; T=0.01/0.01 11:50:12
```

Note: A highly experienced RACF administrator with extensive knowledge of VMSES/E might set up RACF profiles and permissions that allow access to the required resources without the OPERATIONS attribute. Because the maintenance of z/VM and its features is of critical importance to the system operation, the recommended approach of using OPERATIONS is the preferred method for most installations.

OPERATOR and SPECIAL

It is sometimes suggested that the OPERATOR ID be granted the SPECIAL attribute. Whoever, this suggestion is not a preferred practice because access to OPERATOR is broad and uncontrolled in most installations. Even with the access controls that are described in 3.2.8, “Role-based access controls and CP privilege classes” on page 37, OPERATOR is visible to users without security management responsibility.

One reason for giving this access is when an operator inadvertently enters the incorrect time when z/VM is IPLed, which results in all IDs on the system becoming revoked. This issue can be mitigated by using the Auto_Warm_IPL feature statement in SYSTEM CONFIG, combined with effective management of the Hardware Management Console (HMC)/SE time-of-day clock to provide accurate time to the LPAR ToD clocks.

Alternatively, z/VM supports the Server Time Protocol (STP) hardware feature, which can provide an integrated solution for time-of-day management across all systems that are running on the IBM LinuxONE server.

Note: As a preferred practice before granting the OPERATOR user the SPECIAL attribute, consider all possible alternatives. Giving OPERATOR the system-SPECIAL attribute can greatly affect the overall security and integrity of your z/VM system.

Revoking IBMUSER

After the new RACF administrators are defined, log off from IBMUSER. Log on to MAINT (assuming that you gave MAINT RACF authority) and complete the product installation.

Because IBMUSER is a well-known user, it might be a target for unauthorized accesses to your system. To prevent further use of the IBMUSER VM, revoke this VM and remove the operations and special attributes (see Example 4-26). Do *not* delete this VM from your system because IBMUSER ran the exec to generate the RACF database and it is now listed as the owner of all the other VMs on the system.

Example 4-26 RACF alter user for IBMUSER

```
link 7VMRAC10 29e 29e rr
RPIMGR031E RESOURCE 7VMRAC10.29E SPECIFIED BY LINK COMMAND NOT FOUND
Ready; T=0.01/0.01 12:00:18
access 29e 1
DMSACP723I L (29E) R/O
Ready; T=0.01/0.01 12:00:23
rac alu ibmuser revoke
Ready; T=0.01/0.01 12:04:01
rac alu ibmuser nospecial
Ready; T=0.01/0.01 12:04:08
```


rac alu ibmuser nooperation
Ready; T=0.01/0.01 12:04:14

Setting RACF options

Now, define which resources are managed by RACF. The following examples are a good starting point:

- ▶ RAC SETROPTS CLASSACT(VMMDISK)
- ▶ RAC SETROPTS CLASSACT(VMRDR)
- ▶ RAC SETROPTS CLASSACT(VMLAN)

Other options include **VMATCH** and **VMSEGMT**. Also, if your CP directory included **LOGONBY** statements, **RPIDIRCT** created profiles in the SURROGAT class provide the same function. You must activate the SURROGAT class for your LOGONBY function to work as it did before.

It also is a good practice make the corresponding updates to the VMXEVENT to tailor this entry to your installation. This configuration avoids RACF calls for resources that are not RACF-protected and avoids wasting CPU cycles and causing RACF contention.

If you are using DirMaint, use VMXEVENT to exempt the DirMaint service machines from access checking. Because of the number of users in this example, we create a script to issue the required commands, as recommended in *Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283. The script is shown at Example 4-27.

Example 4-27 VMXEVENT EXEC

```
/* REXX */
Parse Upper Arg ID .
If ID = '' Then Do
  Say "Please enter an ID!"
  Exit 1
End
Say "ID" ID "will have a VMXEVENT profile created and populated,"
Say "Enter 'y' to continue:"
Parse Upper Pull Reply
If Left(Reply,1)="Y" Then Do
  Address CMS
  "RAC RDEFINE VMXEVENT USERSEL." || ID
  "RAC RALTER VMXEVENT USERSEL." || ID || " ADDMEM(LINK/NOCTL)"
  "RAC RALTER VMXEVENT USERSEL." || ID || " ADDMEM(STORE.C/NOCTL)"
  "RAC RALTER VMXEVENT USERSEL." || ID || " ADDMEM(TAG/NOCTL)"
  "RAC RALTER VMXEVENT USERSEL." || ID || " ADDMEM(TRANSFER.D/NOCTL)"
  "RAC RALTER VMXEVENT USERSEL." || ID || " ADDMEM(TRANSFER.G/NOCTL)"
  "RAC RALTER VMXEVENT USERSEL." || ID || " ADDMEM(TRSOURCE/NOCTL)"
  "RAC RALTER VMXEVENT USERSEL." || ID || " ADDMEM(DIAG0D4/NOCTL)"
  "RAC RALTER VMXEVENT USERSEL." || ID || " ADDMEM(DIAG0E4/NOCTL)"
  "RAC RALTER VMXEVENT USERSEL." || ID || " ADDMEM(RSTDSEG/NOCTL)"
  "RAC RALTER VMXEVENT USERSEL." || ID || " ADDMEM(MDISK/NOCTL)"
  "RAC RALTER VMXEVENT USERSEL." || ID || " ADDMEM(APPCPWVL/NOCTL)"
  Say "Complete."
End
```

We ran **VMXEVENT EXEC** once for each DirMaint user to be exempted. After the profiles are created for all users, you can activate the VMXEVENT class by using **RAC SETROPTS CLASSACT(VMXEVENT)**.

We also issued **RAC SETEVENT REFRESH USERSEL.xxxxxxxx** commands for the logged-on DirMaint service machines (replacing xxxxxxxx for each DirMaint service machine in turn) for the VMXEVENT definition to take place immediately.

Using group permissions rather than exemption

Depending on your security policy, it might not be appropriate to exempt the DirMaint users from checking. Similar to **RPIDIRECT**, a sufficiently experienced RACF administrator can define resource profiles with the correct access lists to avoid having to use exemption. This task is even easier by using group-based permissions. For example, a group called \$DIRMSRV can be granted appropriate permissions over all the DirMaint server resources, and the DirMaint server user IDs are connected to that group.

This task requires experience in RACF and DirMaint administration, and must not be taken lightly. Follow the installation instructions for DirMaint and use the documented method to exempt the DirMaint users from checking.

Note: The **RAC SETEVENT REFRESH** command must be run on the z/VM system where the user is logged on to take effect. If the user is not logged on, or is logged on to a different member of the SSI cluster, an error message is received, as shown in the following example:

```
RPIS133E SETEVENT FAILED. USER IS NOT CURRENTLY LOGGED ON.
```

To complete the refresh, log on to each system in your cluster and issue the refresh for the users who are logged on to that system. Alternatively, restart the affected servers (which can be done remotely from a central system by running the **AT** command).

4.2.4 Placing RACF into production

Important: This step must be repeated for each member of an SSI cluster.

Run **PUT2PROD EXEC** from the MAINT710 VM.

Note: Make sure that you run the **PUT2PROD EXEC** without any parameters because the \$SERVICE PROD file on the MAINT710 191 disk already lists the components that must be put into production:

```
SERVICE $PRODS A1 V 80 Trunc=80
0 * * * Top of File * * *
1 SERVP2P RACF
2 SERVP2P CP
3 * * * End of File * * *
```

When **PUT2PROD** completes, run **vmfview put2prod** to verify that everything was successful.

Setting up AUTOLOG1 and AUTOLOG2

When doing a normal warm start, the IPL process starts the AUTOLOG1 VM. This process is intended to start the VMs that run in your z/VM environment.

With RACF in place, it is important to ensure that no VMs are started before RACF is properly initialized. RACF provides an AUTOLOG2 user to accommodate this task.

AUTOLOG1 is changed to start only your ESM (RACFVM). After the RACF environment is initialized, RACF runs the **xauto1log** command for the AUTOLOG2 VM, which starts the remaining servers for the system.

The **PROFILE EXEC** for the AUTOLOG1 VM works perfectly for the AUTOLOG2 VM. Therefore, you can copy it to the appropriate disk. Then, you must modify **PROFILE EXEC** for AUTOLOG1 to start only the production ESM (RACFVM), as shown in Figure 4-5.

```
link autolog1 191 11 mr
Ready; T=0.01/0.01 12:25:07
link autolog2 191 12 mr
Ready; T=0.01/0.01 12:25:14
ac 11 x
Ready; T=0.01/0.01 12:25:18
acc 12 z
DMSACC724I 012 replaces Z (011)
Ready; T=0.01/0.01 12:25:29
copy profile exec x = z
Ready; T=0.01/0.01 12:25:53

PROFILE EXEC      X1 V 130 Trunc=130 Size=7
===>
 0 * * * Top of File * * *
 1 /*****
 2 /* Autolog1 Profile Exec */
 3 /*****
 4 'CP XAUTOLOG RACFVM'
 5 'CP LOGOFF'
 6 * * * End of File * * *
```

Figure 4-5 Set up the AUTOLOG1 and AUTOLOG2 virtual machines

You should be able to perform an IPL from the CF1 disk (extent 1) and run in production mode.

4.2.5 Using HCPRWAC

Initially, the system is built with non-aggressive authorization checking with the security parameters in the SYSSEC macro. In fact, most of the entries specify the keyword *defer*, which means that if the ESM does not know what to do with a request, the request is routed to the system CP for determination. What this looks like in the text of the SYSSEC macro is shown in Figure 4-6.

```
HCPRWA  RPIBASE0  E1  F 80          3 Blks 10/25/06 Line    118 of
===>
      SYSSEC ,
          DISKP=ALLOW,DISKU=DEFER,DISKF=FAIL,DISKW=DEFER,DISKM=ON,X
          RDRP=ALLOW,RDRU=DEFER,RDRF=FAIL,RDRW=DEFER,RDRM=ON,    X
          NODEP=ALLOW,NODEU=DEFER,NODEF=FAIL,NODEW=DEFER,NODEM=ON,X
          CMDP=ALLOW,CMDU=DEFER,CMDF=FAIL,CMDW=DEFER,CMDM=ON,    X
          LANP=ALLOW,LANU=DEFER,LANF=FAIL,LANW=DEFER,LANM=ON,    X
          DEFLT=ALLOW,DEFLTU=DEFER,DEFLT=FAIL,DEFLT=DEFER @L2C
      SPACE 3
```

Figure 4-6 HCPRWA assemble file

This model is not a secure model to run the production system. For this reason, after everything is working correctly, change the SYSSEC macro to *fail* instead of *defer*.

In the past, this effort required updating the text of the SYSSEC macro so that it looked like Figure 4-7 and reassembled HCPRWA.

```

HCPRWA  RBOL0001 E1  F 80  Trunc=80 Size=137 Line=120 Col=1 Alt=2
====>
120      SYSSEC ,
121          DISKP=ALLOW,DISKU=FAIL,DISKF=FAIL,DISKW=FAIL,DISKM=ON, X
122          RDRP=ALLOW,RDRU=FAIL,RDRF=FAIL,RDRW=FAIL,RDRM=ON, X
123          NODEP=ALLOW,NODEU=FAIL,NODEF=FAIL,NODEW=FAIL,NODEM=ON, X
124          CMDP=ALLOW,CMDU=FAIL,CMDF=FAIL,CMDW=FAIL,CMDM=ON, X
125          LANP=ALLOW,LANU=FAIL,LANF=FAIL,LANW=FAIL,LANM=ON X
126          DEFLTP=ALLOW,DEFLTU=DEFER,DEFLTf=FAIL,DEFLTW=DEFER @L2C
      SPACE 3

```

Figure 4-7 Modified HCPRWA assembler

To allow clients that do not have HLASM to move to a more secure configuration, the RACF product includes a pre-assembled version of HCPRWA that contains these changes (among others), known as HCPRWAC. For more information about how to use HCPRWAC see Appendix D, “Using HCPRWAC”, in *Secure Configuration Guide for z/VM*, SC24-6323.

HCPRWAC is the IBM-provided modification of HCPRWA that complies with the requirements of LSPP. This process uses VMFUPDAT to update VM SYSSUF.

Complete the following steps:

1. Run the `vmfupdat syssuf` command. Scroll through the panels until you see the Compname for RACF (see Example 4-28).

Example 4-28 MFUPDAT SYSSUF

Update any PPF/component name or YES|NO field. To change all occurrences of a PPF name in the table replace both ***** fields with PPF names.

Compname		Prodid	Servlev	Prodlev	Description

AVS		7VMAVS10	000-0000	000-0000	AVS for z/VM 7.1.0
:INSTALL	YES	:INSPPF	SERVP2P	AVS	
:BUILD	YES	:BLDPPF	SERVP2P	AVS	
:INCLUDE	YES	:P2PPPF	SERVP2P	AVSP2P	
CMS		7VMCMS10	RSU-1901	RSU-1901	CMS for z/VM 7.1.0
:INSTALL	YES	:INSPPF	SERVP2P	CMS	
:BUILD	YES	:BLDPPF	SERVP2P	CMS	
:INCLUDE	YES	:P2PPPF	SERVP2P	CMSP2P	
CP		7VMCPR10	RSU-1901	RSU-1901	CP for z/VM 7.1.0
:INSTALL	YES	:INSPPF	SERVP2P	CP	
:BUILD	YES	:BLDPPF	SERVP2P	CP	
:INCLUDE	YES	:P2PPPF	SERVP2P	CPP2P	

Change PPF name ***** to *****

2. Add the following statement in the file that is shown in Example 4-28:

```
:INCLUDE CCC :P2PPPF SERVP2P RACFP2P
```

Note: Since z/VM 6.3, the HLASM is no longer required to assemble HCPRWA.

3. After you modify the entry for INCLUDE from YES to CCC, select PF5 to process. A flag is raised in the VM SYSSUF file that indicates that RACF was updated and to set this product to BUILD (see Example 4-29). The CPLOAD MODULE is built with the new HCPRWA file (which is the HCPRWAC file). This process changes the parameters from *defer* to *fail*.

Example 4-29 Notification that RACF was updated

```

00037 :PRODID.5684042J%ICKDSF :SERVLEV.RSU-1802 :DESC.ICKDSF DEVICE SUPPORT F
00038 :INCLUDE.YES :INSTALL.YES :INSPPF.SERV2P ICKDSF :BUILD.YES :BLDPPF.SER
00039 ICKDSFP2P :PRODLEV.RDBKZVM1.RSU-1802 RDBKZVM2.RSU-1802 RDBKZVM3.RSU-180
00040 :PRODID.7VMDIR10%DIRM :SERVLEV.000-0000 :DESC.Install/service DirMaint
00041 :INSTALL.YES :INSPPF.SERV2P DIRM :BUILD.YES :BLDPPF.SERV2P DIRM :P2PP
00042 :PRODLEV.RDBKZVM1.000-0000 RDBKZVM2.000-0000 RDBKZVM3.000-0000 RDBKZVM4
00043 :PRODID.7VMRAC10%ACF :SERVLEV.000-0000 :DESC.RACF Feature of z/VM, FL7
00044 :INSPPF.SERV2P RACF :BUILD.YES :BLDPPF.SERV2P RACF :P2PPPF.SERV2P R
00045 :PRODLEV.RDBKZVM1.000-0000 RDBKZVM2.000-0000 RDBKZVM3.000-0000 RDBKZVM4
00046 :PRODID.7VMPTK10%PERFTK :SERVLEV.RSU-1901 :DESC.Performance Tool Kit :I
00047 :INSPPF.SERV2P PERFTK :BUILD.YES :BLDPPF.SERV2P PERFTK :P2PPPF.SERV2
00048 :PRODLEV.RDBKZVM1.RSU-1901 RDBKZVM3.RSU-1901 RDBKZVM4.RSU-1901 RDBKZVM2
00049 :PRODID.7VMHCD10%VMHCD :SERVLEV.000-0000 :DESC.VMHCD for z/VM 7.1.0 :IN
00050 :INSPPF.SERV2P VMHCD :BUILD.YES :BLDPPF.SERV2P VMHCD :P2PPPF.SERV2P
00051 :PRODLEV.RDBKZVM1.000-0000 RDBKZVM2.000-0000 RDBKZVM3.000-0000 RDBKZVM4

```

4. Force the building of the CP nucleus by running the following commands:

```

vmfsetup 7VMRAC10 racf (link
vmfrepl rpiblcprn exec 7VMRAC10 racf (nocopy $select
vmfsetup detach

```

The VMFREPL EXEC is used to support the local modification of replacement maintained parts. VMFREPL can be used to accomplish the following tasks:

- Copy the highest level of a part.
- Copy a specified part.
- Update a Version Vector Table.
- Update a Select Data file.
- Display the highest levels of a part.

RPIBLCPN EXEC is used to build the CPLOAD MODULE by using the RACF files and the version vector tables for RACF. The **\$SELECT** operand adds an entry to the 7VMRAC10 \$SELECT file (see Example 4-30) on the RACFVMs apply disk (2A6), which defines to VMSES/E that local service to the RPIBLCPN EXEC exists.

Example 4-30 7VMRAC10 \$SELECT file

```

7VMRAC10 $SELECT F1 V 80 Trunc=80 Size=2
==>
0 * * * Top of File * * *
1 :APPLYID.07/01/16 09:09:18
2 RPIBLCPN EXC EXC00000 BASE-FILETYPE
3 * * * End of File * * *

```

5. The **SERVICE EXEC** is used again, similar to when you enabled the RACF product. This time, use the **BUILD** operand to create the CPLOAD MODULE by running the following command:

```
service racf build
```

The new CP nucleus, with the RACF CP parts, is placed on the secondary parm disk (default disk address of CF2). For your information, a copy of the previous (or currently running) CPLOAD MODULE is still on the primary (CF1) and tertiary (CF3) parm disks as CPLOAD MODULE. It is also saved on the secondary parm disk as CPLOLD.

6. Shut down the running system.
7. Perform an IPL from the MAINT710 CF2 parm disk.
8. Start the system with the **NOAUTOLOG** parameter.
9. Run **XAUTOLOG RACMAINT**.
10. Run the **PUT2PROD EXEC** from the MAINT VM.

The RACF product for z/VM 7.1.0 is now installed and configured.

4.3 RACF management processes

This section describes how to make DirMaint and RACF work together and shows some basic setup in RACF to protect commonly used resources.

4.3.1 DirMaint changes to work with RACF

The DirMaint-RACF connector provides DirMaint exits that allow the DIRMAINT VM to run the appropriate RACF commands to perform the following tasks:

- ▶ Add a user
- ▶ Define MDISK
- ▶ Define VMRDR
- ▶ Define VMPOSIX
- ▶ Define SURROGAT
- ▶ Define VMBATCH

Note: The DirMaint-RACF connector is one of the reasons that you use DirMaint with your RACF environment. Although it is fairly easy to write your own execs to provide a similar function, the connector is a maintained component of DirMaint.

The code to use this process is included with the base system as part of DirMaint. To implement this process, you update your DirMaint configuration (CONFIGxx DATADVH) with the statements that are defined in the CONFIGRC SAMPDVH file (see Figure 4-8 on page 89).

A copy of the CONFIGRC SAMPDVH file does *not* exist on the DIRMAINT minidisks. Instead, it is on the 2C2 disk that is owned by the 7VMDIR10 VM. In this example, we run **VMLINK** to access this disk and then copy the CONFIGRC SAMPDVH file to our A disk.

Note: The following VMLINK command is used:

```
VMLINK 7VMDIR10 2C2 (FILEL CONFIGRC *
```

This command made it easy to run **COPYF11e** to copy the file and give it the name CONFIGRC DATADVH A.

Complete the following steps:

1. Copy the CONFIGRC SAMPDVH file to your A disk as CONFIGRC DATADVH.

An excerpt from the CONFIGRC DATADVH file is shown in Figure 4-8.

```
CONFIGRC DATADVH A2 V 80 Trunc=80 Size=174 Line=117 Col=1 Alt=0
===>
117 /*! Command handler for DASD Change related commands. */
118 /*!-----*/
119 /USE_RACF= YES DVHRDN EXEC
120 /USE_RACF= NO DVHRDN EXEC
121 ----- 5 line(s) not displayed -----
126 RACF_ADDUSER_DEFAULTS= UACC(NONE)
127 RACF_DISK_OWNER_ACCESS= ACC(ALTER)
128 RACF_RDEFINE_VMPOXIX_POSIXOPT.QUERYDB= UACC(READ)
129 RACF_RDEFINE_VMPOXIX_POSIXOPT.SETIDS= UACC(NONE)
130 RACF_RDEFINE_SURROGAT_DEFAULTS= UACC(NONE) AUDIT(FAILURES(READ))
131 RACF_RDEFINE_VMBATCH_DEFAULTS= UACC(NONE) AUDIT(FAILURES(READ))
132 RACF_RDEFINE_VMRDR_DEFAULTS= UACC(NONE) AUDIT(FAILURES(READ))
133 RACF_RDEFINE_VMMDISK_DEFAULTS= UACC(NONE) AUDIT(FAILURES(READ))
134 ----- 10 line(s) not displayed -----
144 TREAT_RAC_RC.4= 0 | 4
145 ----- 4 line(s) not displayed -----
149 PW_WARN_MODE= MANUAL
150 PW_LOCK_MODE= MANUAL
151 ----- 9 line(s) not displayed -----
160 ESM_PASSWORD_AUTHENTICATION_EXIT= DVHXPA EXEC
161 ----- 14 line(s) not displayed -----
```

Figure 4-8 Part of CONFIGRC DATADVH

The activation of the function for all supported operations is done by using the following line:

```
USE_RACF= YES ALL
```

2. The operation of the function can be altered by changing the parameters in the file. If no changes are necessary, it can be used as is. Run the DirMaint **file** command to store a copy of the CONFIGRC DATADVH file.

Important: This file does not exist on the DIRMAINT user; therefore, specify the file mode on the DirMaint **file** command. Other DirMaint CONFIGxx DATADVH files are on DIRMAINT's D disk, so they can be stored using the following command:

```
DIRM FILE CONFIGRC DATADVH A = = D
```

3. Complete the activation of the connector by running the DirMaint commands to refresh data and configuration files (**DIRM RLDD** and **DIRM RLDC**). You also must give the DIRMAINT and DATAMOVE VMs the RACF special attribute (see Example 4-31).

Example 4-31 RACF authorization for DIRMAINT and DATAMOVE

```
rac alu dirmaint special
```

```
Ready; T=0.01/0.01 11:47:25
```

```
rac alu datamove special
```

```
Ready; T=0.01/0.01 11:47:33
```

After completing this process, when you add a user or minidisk with DirMaint, it is added automatically to the RACF database. For more information, see “Adding virtual machines with DirMaint” on page 90.

4.3.2 RACF authorization concepts

Resources are defined to RACF/VM as profiles in the RACF database. Profiles are available for all of the resources that are defined to a RACF enabled z/VM system (vmmdisk, vmrdr, vmlan, and so on). These profiles can be generic (MAINT.19*, where the asterisk is one or more characters) or discrete (MAINT.CF1), as shown in Example 4-32.

Example 4-32 Discrete and generic profiles

Discrete profiles

```
RDEFINE VMMDISK MAINT.CF1 OWNER(MAINT) UACC(NONE)
PERMIT MAINT.CF1 CLASS(VMMDISK) RESET ID(MAINT) AC(ALTER)
RDEFINE VMMDISK MAINT.CF2 OWNER(MAINT) UACC(NONE)
PERMIT MAINT.CF2 CLASS(VMMDISK) RESET ID(MAINT) AC(ALTER)
...
```

Generic profiles

```
RDEFINE VMMDISK MAINT.CF* OWNER(MAINT) UACC(NONE)
PERMIT MAINT.CF* CLASS(VMMDISK) RESET ID(MAINT) AC(ALTER)
RDEFINE VMMDISK MAINT.190 OWNER(MAINT) UACC(NONE)
PERMIT MAINT.190 CLASS(VMMDISK) RESET ID(MAINT) AC(ALTER)
RDEFINE VMMDISK MAINT.19E OWNER(MAINT) UACC(NONE)
PERMIT MAINT.19E CLASS(VMMDISK) RESET ID(MAINT) AC(ALTER)
...
```

The **RPIDIRCT EXEC** that was used to create the commands to define the RACF database during the installation and configuration process that was used discrete profiles. Your installation must determine whether you want to continue with this practice or use generic profiles. Both methods or a combination of methods work. Make sure that you run **SETROPTS GENERIC(VMMDISK)** before you define the generic profiles.

4.3.3 Adding virtual machines and resources to the system and RACF database

This section describes how to add VMs and resources to the system and RACF database.

Adding virtual machines with DirMaint

This example uses DirMaint as the tool to add VMs to the system. The use of DirMaint allows you to use the DirMaint-RACF connector.

Complete the following steps:

1. When you must add a VM to the system, first make sure that the VM was not defined (see Example 4-33).

Example 4-33 Verifying a virtual machine

```
rac lu userbob
ICH30001I UNABLE TO LOCATE USER    ENTRY USERBOB
Ready(00004); T=0.01/0.01 08:43:53
dirm for userbob get nolock
DVHXTM1191I Your GET request has been sent for processing.
Ready; T=0.03/0.03 08:44:09
DVHREQ2288I Your GET request for USERBOB at * has been accepted.
DVHBDG6209E Specified user USERBOB does not exist, request GET failed.
DVHGET3212E Unexpected RC= 6209, from: EXEC DVHBBGT USERBOB DIRECT A0
DVHREQ2289E Your GET request for USERBOB at * has failed; with RC =
DVHREQ2289E 3212.
```

2. To create a VM, create a file on the A disk of a DirMaint administrator, which contains new VM definition (see Figure 4-9).

```
USERBOB DIRECT    A0  F 80  Trunc=72 Size=5 Line=0
====>
0 * * * Top of File * * *
1 USER USERBOB TEXAS  32m  100m  BCDG
2   INCLUDE IBMDFLT
3   IPL CMS PARM AUTO CR
4   MACHINE XA
5   LINK TCPMAINT 0592 0592 RR
6 * * * End of File * * *
```

Figure 4-9 USERBOB DIRECT

3. Run the command **dirm add** to display a panel, as shown in Example 4-34.

Example 4-34 DIRMAINT ADD

```
-----DirMaint ADD-----
Add a new directory entry for a new USERID, PROFILE, SUBCONFIG, or IDENTITY.
Fill in the USERID, PROFILE, SUBCONFIG, or IDENTITY being added:
====> userbob
Optionally fill in the following when using a prototype:
LIKE ==>          (file name of prototype)
PW   ==>          (password for new user)
VPW  ==>          (password again for verification)
ACCT ==>          (account value for new user - optional)
BUILD ON ==>      (SSI node)
IN   ==>          (identity)
Notes:
- If a value is given for any one of PW, VPW, or ACCT,
  then a value is required for LIKE.
- If a value is given for either PW or VPW,
  then a value is required for both of them.
- BUILD and IN fields can be used for subconfigs only.
```

- If a value is given for either BUILD or IN then a value is required for both of them

5741-A09 (c) Copyright IBM Corporation 1979, 2018.

1= Help 2= Prefix Operands 3= Quit 5=Submit 12=Cursor

4. After entering the name of the VM, press PF5. You receive the messages that are shown in Example 4-35.

Example 4-35 DirMaint Output

```

PUN FILE 0013 SENT TO   DIRMAINT RDR AS  0037 RECS 0013 CPY  001 0 NOHOLD NOKEEP
DVHXMT1191I Your ADD request has been sent for processing to DIRMAINT at
DVHXMT1191I ITS0ZVM1.
Ready; T=0.07/0.08 08:51:11
DVHREQ2288I Your ADD request for USERBOB at * has been accepted.
DVHBIU3450I The source for directory entry USERBOB has been updated.
DVHBIU3424I The next ONLINE will take place immediately.
DVHDRC3451I The next ONLINE will take place via delta object directory.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DMVCTL request has been relayed for processing.
DVHRLA3891I Your DMVCTL request has been relayed for processing.
DVHBIU3428I Changes made to directory entry USERBOB have been placed
DVHBIU3428I online.
DVHBIU3450I The source for directory entry USERBOB has been updated.
DVHBIU3424I The next ONLINE will take place immediately.
DVHDRC3451I The next ONLINE will take place via delta object directory.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DSATCTL request has been relayed for processing.
DVHRLA3891I Your DMVCTL request has been relayed for processing.
DVHRLA3891I Your DMVCTL request has been relayed for processing.
DVHBIU3428I Changes made to directory entry USERBOB have been placed
DVHBIU3428I online.
DVHREQ2289I Your ADD request for USERBOB at * has completed; with RC
DVHREQ2289I = 0.

```

If you run **rac lu** and **dirm for userbob get no lock**, you find that the VM is defined. The **rac lu output** is shown in Example 4-36.

Example 4-36 RACF List User (RAC LU) command output

```

rac lu userbob
USER=USERBOB NAME=UNKNOWN OWNER=DIRMAINT CREATED=19.179
DEFAULT-GROUP=SYS1   PASSDATE=00.000 PASS-INTERVAL= 30 PASSPHRASEDATE=N/A
ATTRIBUTES=NONE
REVOKE DATE=NONE   RESUME DATE=NONE
LAST-ACCESS=UNKNOWN
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED   (DAYS)           (TIME)
-----
ANYDAY                               ANYTIME

```

```

GROUP=SYS1      AUTH=USE      CONNECT-OWNER=DIRMAINT  CONNECT-DATE=19.179
CONNECTS= 00  UACC=NONE      LAST-CONNECT=UNKNOWN
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
  NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
Ready; T=0.01/0.01 08:54:10

```

When you add a minidisk to this user, the minidisk address is added to the RACF database as well. In this example, we ran the **rac rlist** command before *and* after adding a minidisk by using the **DIRM AMD** command. The results are shown in Example 4-37.

Example 4-37 RACF profile added for a DirMaint added minidisk

```

rac rlist vmmdisk userbob.191 auth
ICH13003I USERBOB.191 NOT FOUND
Ready(00004); T=0.01/0.01 08:54:49
. . .
<added minidisk using DirMaint AMDISK command>
. . .
rac rlist vmmdisk userbob.191 auth
CLASS      NAME
-----
VMMDISK    USERBOB.191

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----
00     USERBOB      NONE              NONE          NO

INSTALLATION DATA
-----
NONE

APPLICATION DATA
-----
NONE

SECLEVEL
-----
NO SECLEVEL

CATEGORIES
-----
NO CATEGORIES

SECLABEL
-----
NO SECLABEL

AUDITING
-----
FAILURES(READ)

NOTIFY

```

```

-----
NO USER TO BE NOTIFIED

USER      ACCESS  ACCESS COUNT
-----
USERBOB   ALTER    000000

      ID      ACCESS  ACCESS COUNT  CLASS                      ENTITY  NAME
-----
NO ENTRIES IN CONDITIONAL ACCESS LIST
Ready; T=0.01/0.01 08:55:18

```

Note: The values for the universal access and audit properties of the created minidisk resource profile are set in the **RACF_RDEFINE_VMMDISK_DEFAULTS** parameter in the CONFIGRC DATADVH file. You can also set the default access level that is given to the owner of a disk by using the **RACF_DISK_OWNER_ACCESS** parameter.

Similar parameters exist for the other resource types that are managed by the DirMaint-RACF connector.

Adding virtual machines without DirMaint

If you decide not to use the DirMaint product on your system, an automated process is available that also updates the RACF database. This method is not as automated as the use of the **dirm add** command, but it might be suitable for your installation.

You use the same processes that you use to build the initial RACF database. The processes are the **RPIDIRECT** and **RPIBLDDS** execs. These processes can be completed from the MAINT VM because MAINT can write to the CP directory (and accesses the USER DIRECT file that is found on the PMAINT 2CC disk).

Complete the following steps:

1. Add the new user to the USER DIRECT file (see Figure 4-10).

```

USER      DIRECT  C1  F 80  Trunc=80 Size=5509 Line=5503 Col=1
====>
5503 *
5504 USER USERBOB 18FUMDIM 32M 100M BCDG
5505     INCLUDE IBMDFLT
5506     IPL CMS PARM AUTOOCR
5507     MACHINE XA
5508     LINK TCPMAINT 0592 0592 RR
5509     MDISK 191 3390 2220 10 ZVMUSR MR  ALL  GO4IT  WHYNOT
5510 * * * End of File * * *

```

Figure 4-10 USER DIRECT on the 2CC disk

2. Put the directory online with the **directxa** command after you copy the directory entry for the new user to the user ID DIRECT A file.

Now, the new VM is added to the system directory. However, if you attempt to log on to the VM, the attempt fails (as shown in Example 4-38) because the VM is not defined in the RACF database and you are no longer deferring the request to CP.

Example 4-38 Log on to USERBOB

logon userbob

HCPLGA050E LOGON unsuccessful--incorrect userid and/or password

Enter one of the following commands:

LOGON userid	(Example: LOGON VMUSER1)
DIAL userid	(Example: DIAL VMUSER2)
MSG userid message	(Example: MSG VMUSER2 GOOD MORNING)
LOGOFF	

3. Update the RACF database with information about this VM. To do so, link and access the 651 disk that is owned by 7VMRAC10. You need this disk because that is where the **RPIDIRCT** and MDisk 505 where **RPIBLDDS** execs are stored.
4. Run the **RPIDIRCT EXEC** against the USERBOB DIRECT file (see Example 4-39) to generate an RPIDIRCT SYSUT1 file.

Example 4-39 Run RPIDIRCT

rpidirct userbob direct

USERBOB DIRECT Filemode defaulted to "*".

Output defaulted to "A" disk.

Default group ID = SYS1.

Would you like to change this default?

Enter Y/N

n

Default group ID = SYS1.

DEFINITION pass begins.....

USER USERBOB XXXXXXXX 32M 100M BCDG

INCLUDE IBMDFLT

MDISK 191 3390 30049 1 ZAOL01 MR READ WRITE MULTIPLE

Missing ACIGROUP for userid USERBOB - Defaulted to SYS1

DEFINITION pass complete - PERMIT command generation begins...

NOTE: This EXEC will "PERMIT" only up to 4 indirect LINKS

processing LINK TCPMAINT 592 592 READ for user USERBOB

*** Cannot PERMIT TCPMAINT 592 for Userid USERBOB - no Minidisk

***** scan ended *****

***** 7 Directory records processed *****

***** RPIDIRCT SYSUT1 CREATED *****

The generated RPIDIRECT SYSUT1 file is shown in Figure 4-11.

```
RPIDIRECT SYSUT1  A1  V 80  Trunc=80 Size=16 Line=0 Col=1 Alt=0
===>
 0 * * * Top of File * * *
 1 ***** USERBOB
 2 *
 3 ADDUSER USERBOB DFLTGRP(SYS1) UACC(NONE) PASSWORD(TEXAS)
 4 RDEFINE VMBATCH USERBOB OWNER(USERBOB) UACC(NONE)
 5 PERMIT USERBOB CLASS(VMBATCH) ACCESS(ALTER) RESET
 6 RDEFINE VMRDR USERBOB UACC(NONE) OWNER(USERBOB)
 7 PERMIT USERBOB CLASS(VMRDR) ID(USERBOB) ACCESS(ALTER) RESET
 8 RDEFINE VMMDISK USERBOB.191 OWNER(USERBOB) UACC(NONE)
 9 PERMIT USERBOB.191 CLASS(VMMDISK) RESET ID(USERBOB) AC(ALTER)
10 *
11 *****
12 *
13 *                      PERMIT DIRECTORY LINKS
14 *
15 *****
16 *
17 * * * End of File * * *
```

Figure 4-11 New RPIDIRECT SYSUT1 file

RPIDIRECT did not process the LINK statement in the user definition (see the Cannot PERMIT message in the output of **RPIDIRECT**, and the missing **PERMIT** for TCPMAINT 592 in the **RPIDIRECT SYSUT1** output). This issue appears to occur because **RPIDIRECT** includes support for resolving indirect minidisk links, which adds the **PERMIT** for the actual resource correctly. It does this by creating an index of all minidisks that are defined in the provided directory listing so that it can dereference indirect links. When working with a full directory, this process works as designed, but it fails with a single user's directory entry.

This issue can be resolved by using one of the following methods:

- Add a directory fragment to the userid **DIRECT** file defining the other user and its minidisk. In this example, we add the following two lines to **USERBOB DIRECT** to have **RPIDIRECT** correctly build the required **PERMIT** command:

```
USER TCPMAINT NOLOG
MDISK 592 3390 1 1 ABC123 MR READ
```

The extent that is defined on this **MDISK** statement was irrelevant; it only needed to be present to allow **RPIDIRECT** to build the required **PERMIT** for **USERBOB**.

- Manually add **PERMIT** commands to the **RPIDIRECT SYSUT1** file in response to any Cannot PERMIT messages.

As a preferred practice, use the latter approach. Although having the utility create the correct **PERMIT** automatically is convenient, the directory fragment that is needed to create this **PERMIT** adds a full set of RACF commands to **RPIDIRECT SYSUT1** for that other user.

You must manually remove all those other commands to run **RPIBLDDS** error-free. The issue is compounded if the user you are adding has links to minidisks of many users. All of the other users and their minidisks must be defined in directory fragments, and many unnecessary commands must be cleaned up from **RPIDIRECT SYSUT1**.

In this example, we manually add the appropriate **PERMIT** to the end of the **RPIDIRECT SYSUT1** file to allow the directory link.

5. Run the **RPIBLDDS** exec by using the new **RPIDIRCT SYSUT1** file and update the RACF database with the commands that are shown in Figure 4-12.

```
rpibldds rpidirect
Processing batch file RPIDIRCT SYSUT1 using "RAC" command interface
*
=> ADDUSER USERBOB DFLTGRP(SYS1) UACC(NONE) PASSWORD(TEXAS)
=> RDEFINE VMBATCH USERBOB OWNER(USERBOB) UACC(NONE)
=> PERMIT USERBOB CLASS(VMBATCH) ACCESS(ALTER) RESET
=> RDEFINE VMRDR USERBOB UACC(NONE) OWNER(USERBOB)
=> PERMIT USERBOB CLASS(VMRDR) ID(USERBOB) ACCESS(ALTER) RESET
=> RDEFINE VMMDISK USERBOB.191 OWNER(USERBOB) UACC(NONE)
=> PERMIT USERBOB.191 CLASS(VMMDISK) RESET ID(USERBOB) AC(ALTER)
=> PERMIT TCPMAINT.592 CLASS(VMMDISK) ID(USERBOB) AC(READ)

Ready; T=0.02/0.03 11:11:45
```

Figure 4-12 Run **RPIBLDDS**

6. Run the **rac 1u** command to check how the user is defined to RACF. In this example, we found that the result looked the same as that achieved by using the DirMaint-RACF connector (shown in Example 4-36 on page 92).

4.3.4 Securing your minidisks with RACF

You can use RACF to control who can link the minidisks by using profiles in the VMMDISK resource class. z/VM calls RACF for an authorization check when a user tries to link another user's minidisk. All devices in z/VM except users and groups are considered to be general resources in RACF. Therefore, defining profiles in RACF for resources other than users and groups is done by using the RACF command **RDEFINE**.

You can protect resources by defining the following profiles:

- ▶ Discrete
- ▶ Generic

Discrete profiles are used to protect explicitly a single resource. For example, if a resource requires special access authorization or unique logging information, you can protect it with a discrete profile, as shown in Example 4-40.

Example 4-40 Protect MDisk 191 with a discrete profile

```
RDEF VMMDISK edi.191 uacc(none) own(edi)
```

Note: IBM provides the DirMaint-RACF connector, which defines RACF profiles whenever a change is made to your directory by using DIRM commands. This process is done if a user is added or deleted, for example, and when adding minidisk definitions to virtual guests. However, the DirMaint-RACF connector creates discrete profiles only, which provides a basic security implementation and makes sure that resources are protected.

However, in many installations, the preferred way to protect resources is by defining generic profiles. Generic profiles must contain one or more generic characters. This method might be the best way to protect all resources of the same type of a certain user by defining only one or two profiles.

Note: Consider the following points:

- ▶ Valid generic characters are %, *, and **.
- ▶ Specify % in profile names to match any single non-blank character on the same position of the resource name.
- ▶ Specify * or ** in the profile name to match more than one single character in the same position of the resource name.

For more information, see Chapter 6, “Defining Resources”, in *z/VM RACF Security Server Security Administrator’s Guide*, SC24-6218.

You also can choose to grant permits to groups rather than users. Defining groups to the RACF database is a way to reflect definitions and permissions to your businesses organizational structure and your security policy. It also gives you more flexibility.

Therefore, you might (as shown in the following examples) insert groups into the **ID** keyword of the RACF **PERMIT** command rather than single user IDs. If a user then newly joins a specific organizational unit (that is, the DBADMN unit), connecting this user ID to the defined group provides the user with all the access rights the user needs to do the work.

For more information about the advantages of the use of this group permission rather than user permissions, see “Advantages of using groups” on page 99.

For more information about RACF group structure and security objectives, see Chapter 2, “Organizing for RACF Implementation”, in *z/VM RACF Security Server Security Administrator’s Guide*, SC24-6311.

Note: If you decide to use generic profiles for class VMMDISK, *genlist* this class. Run the following command:

```
SETROPTS GENLIST (VMMDISK)
```

This command causes one copy of each generic profile for the VMMDISK class to be kept in the RACFVM service machine. Changes that are made to generic minidisk profiles are not reflected until a **SETROPTS** refresh command is issued, as shown in the following example:

```
SETROPTS GENERIC (VMMDISK) REFRESH
```

Linux for IBM System z® guests often must have many database volumes that are attached to their VMs and volumes that are of the same kind and protection level needs. This configuration can be done by defining profiles, as shown in Example 4-41 and Example 4-42.

Example 4-41 Protect database volumes with a generic profile

```
RDEF VMMDISK LNX1.* UACC(NONE) OW(LINX1)
```

Example 4-42 Permit access to a database volume by using a generic profile group access list

```
PERMIT LNX1.* CLASS (VMMDISK) ID (DBADMN) ACC (UPDATE)
```

If your system MDisk should not be updated by **DBADMN** but by **SYSPROGS** group, and the virtual addresses of these types of minidisks start with 20, use the commands that are shown in Example 4-43 on page 99.

Example 4-43 Generic profile and group permission example

```
RDEFINE VMMDISK LNX1.20* uacc(none) ow(LINX1)
PERMIT LNX1.20* CLASS(VMMDISK) ID(SYSPROG) ACC(UPDATE)
```

Advantages of using generic profiles

The use of generic profiles includes the following advantages:

- ▶ The number of profiles in the RACF database is reduced significantly.
- ▶ No RACF administrator action is needed when MDiskS are added or removed.
- ▶ The principle of least privilege is met.
- ▶ A good overview of the security setup in your RACF database occurs, which might be helpful when showing security concepts to auditors.

This principle can be used for almost all of the general resource profiles except VMLAN VSWITCH devices.

Advantages of using groups

The use of groups includes the following advantages:

- ▶ Access lists in resource profiles include fewer entries.
- ▶ Changes in your companies business organization can be easier reflected in permission structures.
- ▶ If people change departments in your organization, accesses are easily withdrawn and granted by removing them from a group and connecting them to another group.
- ▶ RACF-DB is provided with a structure that is adopted to your business needs.
- ▶ They are a good overview of the security setup in your RACF database.

4.3.5 Securing guest LANs and virtual switches with RACF

RACF can be used to protect VLANs and virtual switches by using profiles in the VMLAN class. After defining the appropriate profiles, be sure to activate your VMLAN class by running the following command:

```
SETRPTS CLASSACT(VMLAN)
```

The VMLAN class contains the following sets of profiles to protect LANs:

- ▶ Base profiles control the ability of a z/VM user to use a LAN.
- ▶ VLAN-ID qualified profiles are used to assign a user to one or more IEEE VLANs.

Base profiles

Base profiles are called *userid.name*, where *userid* is the LAN owner and *name* is the name of the LAN. Both qualifiers are a maximum of 8 characters. In the case of a VSWITCH, *userid* is always SYSTEM. These profiles control the authorization and auditing of attempts by any user to **COUPLE** to a guest LAN of virtual switch. A user must have UPDATE access to the profile to be authorized for the **COUPLE** command.

VLAN-ID qualified profiles

Two types of virtual switches are available: user-based and port-based. The default is user-based. Access to the virtual switch is on a user ID basis. All ports for a guest have the same attributes and VLAN IDs.

If a virtual switch is VLAN-aware (which is done by setting the **VLAN defvid** parameter), a secondary set of VLAN ID-qualified VMLAN profiles are used to control the ability of user IDs to connect to a particular IEEE VLAN. Profiles of this type are named `SYSTEM.name.vid`, where *name* is the name of the virtual switch and *vid* is a VLAN ID of the value 1 - 4096, inclusive. In this case, the *vid* must consist of 4 digits.

A user who wants to connect to a virtual switch of this type must have UPDATE access to the qualified profile. The VLAN-Id qualified profiles are checked only if the user has UPDATE access to the base profile protecting the virtual switch.

Note: Consider the following points:

- ▶ VLAN-Id qualified profiles must be discrete; generic profiles are ignored.
- ▶ Global access checking cannot be used for VLAN ID-qualified profiles.

For more information about protecting VLAN resources, see Chapter 10, “Protecting z/VM Resources”, of *z/VM RACF Security Server Security Administrator's Guide*, SC24-6218.

Base profiles control the ability of a guest to connect to the LAN (automatically through the directory **NICDEF** statement or by using the CP **COUPLE** command), and VLAN-ID qualified profiles to control access to specific VLANs on IEEE VLAN-aware virtual switches. To couple to a guest LAN or a virtual switch, a user must have UPDATE access to the profile.

For example, to control access to guest LAN NET100, which is owned by klausm, you must define the profile that is shown in Example 4-44.

Example 4-44 Authorize virtual guest LNX01 to couple to a LAN

```
RDEF VMLAN EDIALVES.NET100 UACC(none) OWNER(EDIALVES)
PERMIT EDIALVES.NET100 CLASS(VMLAN) ID(LNX01) ACC(UPDATE)
```

For more information about guest LANs and virtual switches, see *z/VM Connectivity* SC24-6267.

For IEEE VLAN-aware virtual switches, the mechanism to get access is much the same, although the profile looks different. For this type of virtual switch, you must define the VLAN-ID in the protecting profile. Profiles of this type are set up as `SYSTEM.name.vid`, where *name* is the name of the virtual switch and *vid* is a VLAN ID having a value 1 - 4094, inclusive.

The *vid* qualifier must consist of four decimal digits, and leading zeros must be entered for VLAN IDs with fewer than 4 digits.

In Example 4-45, a user-based virtual switch named VSWINT (virtual switch for internal use only) is defined. Also, the VLAN IDs 10 and 20 are assigned to different user IDs (which can be group IDs). In this example, VLAN ID 10 is used by the segment NETADMINS and VLAN ID 20 is used by LNXBANK (Linux guests for running a banking application).

Example 4-45 Defining the base profile

```
RDEFINE VMLAN SYSTEM.VSWINT UACC(NONE)
PERMIT SYSTEM.VSWINT CLASS(VMLAN) ID(NETADMN LNXBANK) ACC(UPDATE)
RDEFINE VMLAN SYSTEM.VSWINT.0010 UACC(NONE)
PERMIT SYSTEM.VSWINT.0010 CLASS(VMLAN) ID(NETADMNS) ACC(UPDATE)
RDEFINE VMLAN SYSTEM.VSWINT.0020 UACC(NONE)
PERMIT SYSTEM.VSWINT.0020 CLASS(VMLAN) ID(LNXBANK) ACC(UPDATE)
```

VLAN accesses can be separated from each other by using this method.

Accessing multiple VLANs from a guest

z/VM Virtual Switch supports access ports (where the guest is VLAN-unaware and the VSWITCH handles all VLAN tagging) and trunk ports (where the guest must be VLAN-aware and process its own VLAN tagging).

Without RACF, access to VLANs is controlled by the **GRANT** option of the CP **SET VSWITCH** command (**MODIFY VSWITCH** in **SYSTEM CONFIG**). For a specific user, a set of VLANs can be granted on a VSWITCH by listing them in the **VLAN** parameter. If more than one VLAN is specified, the **PORTTYPE** parameter must also be set to **TRUNK**. If a list of VLANs is given but **PORTTYPE ACCESS** is used, an error occurs, as shown in Example 4-46.

Example 4-46 SET VSWITCH GRANT with multiple VLANs and PORTTYPE ACCESS

```
set vswitch vlantst grant tcpip vlan 10 20 30
HCPSWS2847E PORTTYPE ACCESS is not allowed when the user is authorized
HCPSWS2847E for more than one VLAN
```

With RACF in place and the VMLAN class active, **SET VSWITCH GRANT** is not used. Instead, when a user network interface card (NIC) attempts to connect to a VSWITCH that is VLAN-aware, CP requests the list of all profiles to which the user has permission. If this list returns more than one VLAN profile, CP treats this the same as multiple VLAN numbers on the **SET VSWITCH GRANT VLAN** option and expects the **PORTTYPE** to be **TRUNK**.

This behavior can create unexpected results when you are using group-based access management, as described in 4.3.4, “Securing your minidisks with RACF” on page 97. You might want all of a set of Linux systems to belong to a particular group for DASD management, for example, but if they attach to different VLANs on a VSWITCH, you cannot use the same group for VLAN management. Use different group structures for different resource types to allow for different access mappings between those resource types.

SYSSEC considerations of guest LANs

The SYSSEC macro, which is coded in the RACF module HCPRWA, can influence the final result of resource requests in the VMLAN class. If a VLAN is not protected by a RACF profile and if RACF is active on the system, SYSSEC can be coded to let RACF perform one of the following tasks:

- ▶ Allow access
- ▶ Deny access
- ▶ Defer access decision to a z/VM

To check the settings of your SYSSEC macro and for more information about the SYSSEC macro, see *z/VM RACF Security Server Macros and Interfaces*, SC24-6309.

4.3.6 Labeled security and mandatory access control

RACF supports the use of security labels, which allows an installation to implement a security policy that employs *mandatory access control* (MAC). Standard RACF profiles and ACLs are a form of *discretionary access control* (DAC), where individual resources are protected explicitly by the ACLs that are defined in their profile. MAC uses security labels to classify users and resources into security zones and classify access to those zones.

Note: z/VM 6.4 with RACF and security labels in place was evaluated against the Common Criteria Labeled Security Protection Profile (LSPP). The evaluated configuration received an Evaluation Assurance Level of EAL4+. z/VM V7.1 is designed to comply with the same Common Criteria requirements as were successfully evaluated for z/VM V6.4.

For more information about the evaluated configuration, see *Secure Configuration Guide for z/VM*, SC24-6323.

Using labeled security

Labeled security is implemented in RACF for z/VM by using the SECLABEL class. *RACF Security Server Security Administrator's Guide*, SC24-6311 describes the use of security labels in achieving a security model employing MAC. In addition, the specific implementation of security labeling that was used in the evaluation of z/VM 7.1 against the LSPP can be found in *Secure Configuration Guide for z/VM*, SC24-6323. Either of these documents provide thorough examples about how to use the SECLABEL class in RACF for MAC.

Labeled security does not provide a more secure system. In fact, it can be argued that the use of MAC alone provides less security over individual resources. The reason this is the case is that MAC does not focus on the specific resources in a configuration, but rather on the categories and zones to which the resources belong. Instead of permissions being granted to specific discrete resources (as happens in a DAC model), permissions are granted across the security zone with MAC.

SECLABEL and Linux virtual machines

In the case of a Linux on z/VM environment, the use of MAC might result in individual Linux VMs accessing a wider set of resources than DAC. For example, suppose that SECLABEL was used to protect the disks that are attached to a set of database server guests. The data that is contained in these databases is assessed as being at the same security level, so all of the database minidisks get the same label applied and the Linux guest IDs are assigned that label. Now, where before in a DAC model each server had access to its own disks only, the use of MAC alone means that any of these servers can access any of the database disks.

Combining DAC and MAC

MAC can bring a higher level of security to a Linux on z/VM configuration. In our example configuration, a set of database servers holds sensitive customer data and another set of database servers with less sensitive data. These database servers are allocated with the appropriate SECLABELs that reflect the different security zones of the data under management.

Now, suppose a malicious system administrator (with sufficient authority to manage discrete resource profiles for the Linux guests) wanted to access sensitive data by using one of the servers with less stringent controls. This operator issues **PERMIT** commands to allow a less secure server to access physically the sensitive data disks. In a system with DAC alone, this change is all that is required for the less secure server to link the disks and access the data.

When MAC is active, the request is rejected regardless of the discrete **PERMIT** commands because the SECLABELs of the servers and minidisks do not allow the access. In this way, SECLABELs provide an extra layer of security protection.

Note: Implement MAC that uses the RACF SECLABEL class as another security protection over and above standard DAC rather than as a security model in its own right.

4.3.7 Backing up the RACF database

The default configuration of RACF provides a primary and a backup database. As supplied, RACFVM uses a data set that is called RACF.DATASET (which is on the virtual device 200) as its primary database, and a data set called RACF.BACKUP (which is on the virtual device 300) as its backup database. Also, RACFVM keeps the backup database up to date with changes that are made to the primary database (except for the recording of statistics). These specifications are set in the RACF database name table (ICHRDSNT). RACF for z/VM comes with a default ICHRDSNT that defines these settings.

The RACF primary and backup databases are accessed from the time RACFVM starts. This configuration allows RACFVM to keep the backup in-step with the primary, and allows the active database to be switched if needed. However, it makes it slightly more difficult to make a copy of the database because a RACF database must be copied only when it is not active.

For most installations, the backup copy of the database as kept by RACFVM might not be sufficient. It does not protect the database from being lost if a disk subsystem is lost or a disaster occurs, for example. Every installation of RACF should implement a method to back up the database, and keep that backup separate from the running system.

Making an extra backup

RACF Security Server System Programmer's Guide, SC24-6312 provides examples about how to use the RACF database utilities IRRUT200 and IRRUT400 to perform backups of RACF databases. This scenario is based on an IRRUT400 example entitled "Copying a RACF database to a larger volume without shutting down the RACFVM server" from *z/VM: RACF Security Server System Programmer's Guide*, SC24-6312.

Note: Make this kind of backup during a period of as little system activity as possible.

Complete the following steps:

1. Log on to the RACMAINT user.
2. Send a message to RACFVM to detach the F200 and F300 disks so that RACMAINT can link to them.
3. Link to the F200 disk of RACFVM (the original supplied RACF database primary disk) as a staging area for the backup.
4. Because IRRUT400 requires system CMS to run, perform an IPL of the CMS saved system.
5. Run RACUT400 to copy the database.

Example 4-47 shows these steps.

Example 4-47 Use IRRUT400 to back up the RACF database

```
SEND CP RACFVM DET F200 F300
Ready; T=0.01/0.01 16:29:45
LINK RACFVM F200 400 W
Ready; T=0.01/0.01 16:29:55
IPL CMS
z/VM V7.1.0    2019-06-13 16:18
DMSACP723I D (192) R/O
DMSACP723I B (305) R/O
DMSACP723I T (190) R/O
DMSACP725I 190 also = S disk
```

Ready; T=0.01/0.01 17:04:33

RACUT400

This exec is used to Split/Merge or Create a copy
of a RACF data base.

Press Enter to continue....

<Enter>

Do you wish to SPLIT a RACF data set into multiple extents?

or

Do you wish to MERGE multiple RACF data sets into 1 or more extents?

or

Do you wish to COPY one RACF data set into another extent?

Enter SPLIT or MERGE or COPY or QUIT

copy

A single Racf Data set is to be copied to another extent.

Enter the single input device address

200

Enter the single output device address

400

DMSACC724I 200 replaces R (200) - OS

DMSACC723I R (0200) R/W - OS

DMSACC724I 400 replaces X (400) - OS

DMSACC723I X (0400) R/W - OS

The following are the Input Racf Data Set(s)

"RACF.DATASET" (vaddr = 200)

The following are the Output Racf Data Set(s)

"RACF.DATASET" (vaddr = 400)

Do you wish to continue?

Enter YES or NO

yes

You will now be prompted for Input Parameters to 'IRRUT400'

A series of panels containing a full description of these Parameters
can be viewed by entering HELP

Enter HELP for a description of input Parameters

or

Enter CONT to continue without the Parameter description

or

Enter QUIT to terminate

cont

Enter Input Parameter one at a time for 'IRRUT400'

or

Enter END to use default values

no!lockinput

Enter Next Parameter for 'IRRUT400'

or

Enter END to specify end of input

or

Enter QUIT to terminate.

end

Processing begins

All output will be placed in the 'UT400 OUTPUT' file on the 'A' disk.

Program 'IRRUT400' is being executed - Please wait -

Processing completes

Return code from 'IRRUT400' = 0

The primary RACF database is copied to the RACFVM F200 disk. You can now perform other operations on this copy, such as reporting or making further backups by using DDR or other facilities.

Using the RACUT200 and RACUT400 tools

The RACUT200 and RACUT400 execs that start the RACF utilities are sensitive to the types of disks that are used. It also makes assumptions about the type of device to expect based on the device addresses used.

In this example, when we attached the F200 minidisk by using F200 as the virtual device address, the device address was rejected by the utility as invalid. Only the common device addresses that are used for RACF database minidisks (200, 300, and 400) are accepted by the tools.

When we attempted to perform the copy that is shown in Example 4-47 on page 103 by using the full-pack minidisk that is attached at 200 and the F200 minidisk that is attached at 300, the utility failed with a message saying the output data set is invalid. It seems that safety checks exist that are built in to the utilities.

If you use the 200 and 300 devices, the utilities seem to treat them as though they should be the pair of RACF primary and backup disks, and check the data set names to be as expected. In our case, the real 200 disk and the F200 disk have the data set name RACF.DATASET, and this issue caused the utility to fail. Attaching the F200 minidisk at 400 instead worked well because the utility makes no assumptions about the name of the data set that should appear at device 400.

4.3.8 RACF recovery options

If a system availability issue occurs, it might be necessary to recover RACF data from a backup. Circumstances also might exist that prevent the RACFVM server from starting.

This section introduces some basic methods to use to perform recovery of RACF.

Note: For more information about the recovery procedures for events that can compromise RACF operation and about RACF recovery or for any other specific scenarios, see Chapter 7, “Recovery procedures” of *z/VM V7.1 RACF Security Server System Programmer's Guide*, SC24-6312.

Recovery of the RACF primary database

If the RACF primary database is unavailable or in error, the following options are available:

- ▶ If the backup database is valid, you can use RACUT200 to copy the valid backup to the primary volume.
- ▶ If no backup exists, you can restore the most recent memory dump of the database by using a DDR or RACF utility.

We describe a scenario in which we must recover the RACF primary database disk from the backup we took by using the procedure that is described in “Making an extra backup” on page 103.

If RACF cannot start

RACF includes an operation mode called *failsoft processing* that it adopts if no primary databases are available. In failsoft processing, if RACF cannot authorize an access request by using in-memory tables, it prompts the operator for a decision on the access request.

Note: For SSI, it is required that the RACF database is shared between the members of an SSI cluster. For more information about DASD sharing, see the Sharing RACF Databases in a z/VM Single System Image Cluster section of *z/VM: RACF Security Server System Programmer's Guide*, SC24-6312, and *z/VM: CP Planning and Administration*, SC24-6271.



Security policy management on IBM z/VM

Most organizations have a security policy that typically states the rules for controlling access to data. Statements for data ownership and rules about granting the least access that is necessary for each role also are in place.

However, few instructions might be available about the practical scenario of implementation. Little mention often is made of any of the IT platforms that are involved. Therefore, little or no link exists between that policy and the security procedures that must exist.

Organizations find a great benefit in having documentation that relates the policy to the platform, and for each software product that needs security-related configuration. It should be possible to see the line from policy to procedures, and see that the policy is enforced in the implementation environment.

This chapter provides an overview of how to implement some of the common statements that are defined on a security policy.

This chapter includes the following topics:

- ▶ 5.1, “User ID management” on page 108
- ▶ 5.2, “Communication encryption” on page 127
- ▶ 5.3, “Single System Image Security” on page 128
- ▶ 5.4, “Auditing” on page 130

5.1 User ID management

User ID management is closely related to how the security policy is described. All of the management of user ID identity, access, and entitlement should be in accordance with each of the policies that are described in the company's security policy.

This section describes some mechanisms that are available on z/VM to control user IDs and their accesses and entitlements.

5.1.1 Least privilege principle

In an operating system, some operations are privileged and the permission to perform these operations are restricted to authorized users. These privileged operations usually include tasks, such as restarting the system, adding and modifying privileges to other users, adding and deleting users, and modifying the system date and time.

A system that is secure requires that each user is granted only those privileges that are necessary to complete their task. Privileges provide the advantage that only users that require certain privileges must be granted these privileges.

This restriction of privileges is known as the *principle of least privilege*, and it is useful in limiting damage to the system that can result from an accident, error, or malicious administrators and operators. It also is useful when the system must be audited. The audit of a privileged task is reduced to those users that are allowed to run that task.

CP privilege classes

As described in 3.2.8, "Role-based access controls and CP privilege classes" on page 37, one of the ways to control privileges for a user is through z/VM privilege classes. Every user that is defined in the z/VM User Directory has one or more privilege classes.

When the system security policy follows the enterprise security policy, privilege classes represent jobs or roles on the system and are associated with an enterprise security policy job or role. The privilege classes are used in z/VM to implement *role-based access control* (RBAC).

Seven privilege classes are defined by default in z/VM, which are represented by A - G. These letters represent the specific roles in the z/VM operating environment, ranging from System Operator to General User. By using the default classes, a privileged user is any user with a class other than class G authority on the system.

It is possible to create privilege classes that meet the enterprise security policy according to the roles that are described in it. These classes can be represented by I - Z, or 1 - 6. Example 5-1 shows changing the **SHUTDOWN** command from privilege class A to privilege class S. In this situation, only users with privilege class S are authorized to shut down the system.

Example 5-1 Changing the SHUTDOWN command to privilege class S

q cpcmd shutdown

Command: SHUTDOWN

Status: Enabled Not Silent

IBM Class: A PrivClasses: A

CMDBK Address: 009EEBF0 Entry Point: HCPSHUTD

Ready;

cp modify command shutdown privclasses s

Ready;

q cpcmd shutdown

Command: SHUTDOWN

Status: Enabled Not Silent
IBM Class: A PrivClasses: S
CMBK Address: 009EEBF0 Entry Point: HCPSHUTD

Command: -----

Status: Enabled Not Silent
IBM Class: A PrivClasses: A
CMBK Address: 01E00020 Entry Point: HCPSHUTD

Ready;

In Example 5-1 on page 108, the privilege class modification was done dynamically. If a restart is done on the system, the change is lost. To make the change permanent, update SYSTEM CONFIG to reflect the changes. The entry in the SYSTEM CONFIG file looks like the following string:

Modify cmd SHUTDOWN ibm A priv A

To determine which classes to which a user has access, run the **QUERY PRIVCLASS** command. To determine what CP commands and diagnostic instructions to which a user has access, run **QUERY COMMANDS**. Example 5-2 shows user EDI privilege class and the commands that are available to it.

Example 5-2 Privilege class and commands that are available for default classes G and ANY

query privclass

Privilege classes for user EDI

Currently: G

Directory: G

Ready; T=0.01/0.01 16:43:49

query commands

ADJUNCT	ADSTOP	ATTN	BEGIN	CHANGE	CLOSE
COMMANDS	COUPLE	CPFORMAT	CPU	DEFINE	DETACH
DIAL	DISCONNECT	DISPLAY	DUMP	ECHO	EXTERNAL
FOR	INDICATE	IPL	LINK	LOADVFCB	LOCATEVM
LOGON	LOGOFF	MESSAGE	NOTREADY	ORDER	PURGE
QUERY	READY	REDEFINE	REQUEST	RESET	RESTART
REWIND	SCREEN	SEND	SET	SIGNAL	SILENTLY
SLEEP	MSG	SPOOL	SPXTAPE	STOP	STORE
SYSTEM	TAG	TERMINAL	TRACE	TRANSFER	UNCUPLE
UNDIAL	VDELETE	VINPUT	VMDUMP	XAUTOLOG	XSPPOOL
DIAG00	DIAG08	DIAG0C	DIAG10	DIAG14	DIAG18
DIAG20	DIAG24	DIAG28	DIAG40	DIAG44	DIAG48
DIAG4C	DIAG54	DIAG58	DIAG5C	DIAG60	DIAG64
DIAG68	DIAG70	DIAG7C	DIAG88	DIAG8C	DIAG90
DIAG94	DIAG98	DIAG9C	DIAGA0	DIAGA4	DIAGA8
DIAGB0	DIAGB4	DIAGB8	DIAGBC	DIAGC8	DIAGD0
DIAGDC	DIAGE0	DIAGE4	DIAGEC	DIAGF0	DIAGF8
DIAG204	DIAG210	DIAG214	DIAG218	DIAG220	DIAG224
DIAG238	DIAG23C	DIAG240	DIAG244	DIAG248	DIAG250
DIAG254	DIAG258	DIAG260	DIAG264	DIAG268	DIAG26C
DIAG270	DIAG274	DIAG278	DIAG27C	DIAG280	DIAG288
DIAG29C	DIAG2A0	DIAG2A4	DIAG2A8	DIAG2C4	DIAG2E0

COMMAND directory statement

In some cases, a user might need to run a privileged command during logon to set up a user, but does not need to have authorization to run all the commands of the privilege class of this command. One of the solutions is to move this command to a new privilege class and grant access to this new class for the user.

Another solution where you do not need to create a privilege class is to place the **COMMAND** statement into the user directory. The **COMMAND** statement is part of a user directory entry. This statement, which supports up to 255 characters, can run a privileged command after the instantiation of a VM but before the guest formally undergoes an IPL. This command bypasses the need to give a user a specific clearance level while allowing flexibility in configuration.

When the **COMMAND** statement is used, make sure it is defined before any device statement and any command operands are specified uppercase. In Example 5-3, user EDI has **QUERY LPAR** specified at its directory. This command is available to only privilege classes B and E. The user can run the command during the logon process.

Example 5-3 Running the COMMAND directory statement

```
USER EDI LNX4ITS0 64M 96M G
  COMMAND QUERY LPAR
  SPOOL 000C 2540 READER *
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
  CONSOLE 009 3215 T

Log on process:

LOGON EDI
z/VM Version 7 Release 1.0, Service Level 1801 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES: 0001 RDR, NO PRT, NO PUN
LOGON AT 14:10:36 EDT TUESDAY 06/11/19
No LPAR data is available
ctive partition: MUSCA13
Ready; T=0.01/0.01 10:04:35
```

Although the **COMMAND** statement is limited to 255 characters, multiple statements can exist for a single user definition.

IBM Resource Access Control Facility optional user attributes

When a system is IBM Resource Access Control Facility (RACF) protected, it is possible to assign attributes to users by running RACF commands. User attributes describe various extraordinary privileges, restrictions, and processing environments that can be assigned to specified users.

It is possible to assign attributes at the system level or group level. When assigned at the system level, attributes are effective for the entire RACF-protected system. When assigned at the group level, their effect is limited to profiles that are within the scope of the group. The scope of control of a group-level attribute is inherited to the group-ownership structure to its subgroups until a subgroup is owned by a user, rather than a superior group.

Figure 5-1 shows an example of how the attributes are inherited through subgroups. As shown in Figure 5-1, GROUP1 owns GROUP2, GROUP2 owns GROUP3 and USER1, and so on. A user who is connected to GROUP1 with the group-SPECIAL attribute has an explicit scope of control as shown in the figure. That is, the user cannot modify any profiles that are owned by GROUP5.

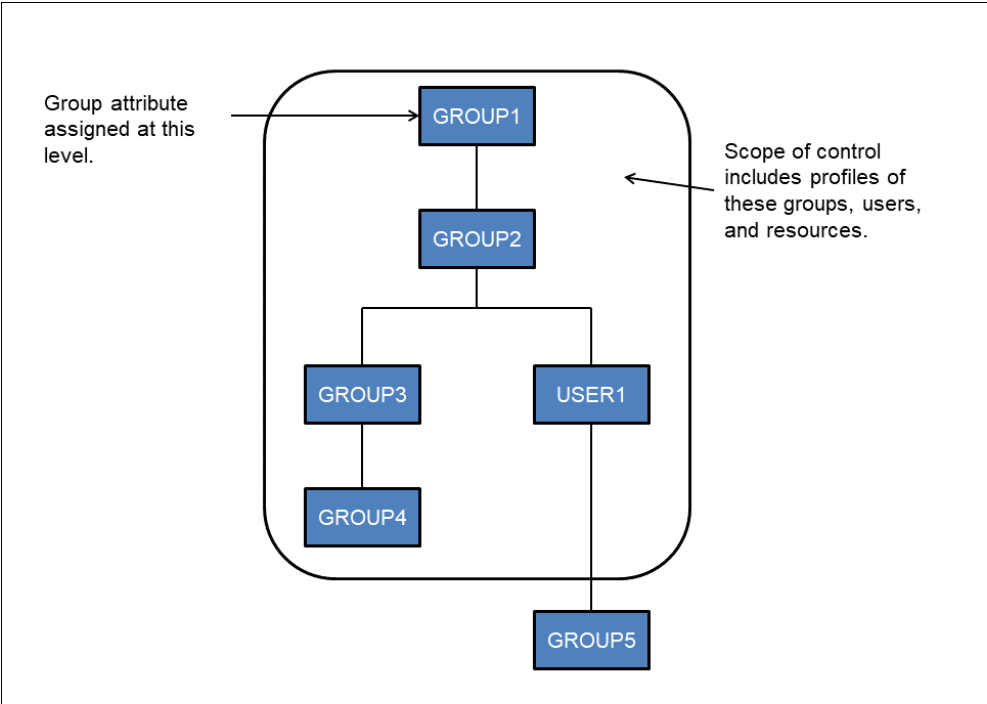


Figure 5-1 Scope of control of an attribute at group level

Following the least privilege principle, SPECIAL, AUDITOR, and OPERATIONS attributes are assigned to a minimum number of people in the system to administer security.

Table 5-1 lists the user attributes that are available in RACF and its descriptions.

Table 5-1 User attributes

User attribute	Description
SPECIAL	The SPECIAL attribute gives the user full control over all the RACF profiles in the RACF database when assigning it at the system level. At the system level, the SPECIAL attribute allows the user to issue all RACF commands. When you assign the SPECIAL attribute at the group level, the group-SPECIAL user has full control over all resources that are within the scope of the group, and cannot issue RACF commands that have a global effect on RACF processing.
AUDITOR	<p>When assigning the AUDITOR attribute at the system level, it gives the user full responsibility for auditing the security controls and the use of system resources across the entire system. With it, the user can specify logging options on the RACF commands, list the auditing options of any profiles by using the RACF commands, and control other logging to SMF for detecting changes and attempts to change the RACF database or for detecting accesses and attempted accesses of RACF-protected resources.</p> <p>When assigning the AUDITOR attribute at the group level (that is, when assigning the group-AUDITOR attribute), authority is restricted to resources that are within the scope of the group.</p>

User attribute	Description
ROAUDIT	The user with the ROAUDIT attribute (read-only auditor) has the same primary responsibility as the user with the AUDITOR attribute; that is to monitor the system. The difference between the AUDITOR attribute and the ROAUDIT attribute is that a user with the AUDITOR attribute can monitor the system and set auditing controls. A user with the ROAUDIT attribute can monitor only the system with the set of auditing controls
OPERATIONS	When assigning this attribute at the system level, it allows the user to perform any maintenance operations, such as copying, reorganizing, cataloging, and scratching, on RACF-protected resources. At the group-OPERATIONS level, the authorization to perform these operations is restricted to the resources that are within the scope of the group.
CLAUTH	The CLAUTH (class authority) attribute allows the user to define profiles in a specific RACF class. A user can have class authority for the USER class and any of the classes that are defined in the class descriptor table (CDT).
REVOKE	This attribute excludes the RACF defined user from entering the system. Revoke can be assigned at the group level, in which case the user cannot enter the system that is connected to that group.
PROTECTED	A protected user ID cannot be used to enter the system by any method that uses a supplied password, such as CP logon, rlogin, or FTP. Also, a protected user ID cannot be revoked through inactivity or unsuccessful attempts to access the system by using an incorrect password or password phrases. A protected user ID is defined by assigning the NOPASSWORD and NOPHRASE attributes through the ADDUSER or ALTUSER command.

To show the attributes of a user, list the user's profile. Example 5-4 show attributes SPECIAL and OPERATIONS that are assigned to user EDI.

Example 5-4 Displaying attributes on a user

```

rac lu edi
USER=EDI NAME=UNKNOWN OWNER=IBMUSER CREATED=19.167
DEFAULT-GROUP=SYS1 PASSDATE=19.167 PASS-INTERVAL= 30 PHRASEDATE=N/A
ATTRIBUTES=SPECIAL OPERATIONS
REVOKE DATE=NONE RESUME DATE=NONE
LAST-ACCESS=19.172/15:14:17
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME

```

To give an attribute to a user, run the **ALTUSER** command. Example 5-5 shows the attribute SPECIAL being added to user EDI.

Example 5-5 Adding attribute SPECIAL to user EDI

```

rac lu edi
USER=EDI NAME=UNKNOWN OWNER=IBMUSER CREATED=19.167
DEFAULT-GROUP=SYS1 PASSDATE=19.167 PASS-INTERVAL= 30 PHRASEDATE=N/A
ATTRIBUTES=NONE
REVOKE DATE=NONE RESUME DATE=NONE
LAST-ACCESS=19.168/15:26:13
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME

```

```

LOGON ALLOWED      (DAYS)          (TIME)
-----
ANYDAY              ANYTIME
GROUP=SYS1          AUTH=USE        CONNECT-OWNER=IBMUSER  CONNECT-DATE=19.167
CONNECTS=          02 UACC=NONE    LAST-CONNECT=19.168/15:26:13
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE   RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
  NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
Ready;

```

rac alu edi special

Ready;

rac lu edi

```

USER=EDI NAME=UNKNOWN OWNER=IBMUSER  CREATED=19.167
DEFAULT-GROUP=SYS1     PASSDATE=19.167 PASS-INTERVAL= 30 PHRASEDATE=N/A
ATTRIBUTES=SPECIAL
REVOKE DATE=NONE   RESUME DATE=NONE
LAST-ACCESS=19.168/15:26:13
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED      (DAYS)          (TIME)
-----
ANYDAY              ANYTIME
GROUP=SYS1          AUTH=USE        CONNECT-OWNER=IBMUSER  CONNECT-DATE=19.167
CONNECTS=          02 UACC=NONE    LAST-CONNECT=19.168/15:26:13
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE   RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
  NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
Ready;

```

To remove an attribute from a user, run the **ALTUSER** command. Example 5-6 shows the removal of the **SPECIAL** attribute from user **EDI**.

Example 5-6 Removing the SPECIAL attribute from user EDI

rac lu edi

```

USER=EDI NAME=UNKNOWN OWNER=IBMUSER  CREATED=19.167
DEFAULT-GROUP=SYS1     PASSDATE=19.167 PASS-INTERVAL= 30 PHRASEDATE=N/A
ATTRIBUTES=SPECIAL
REVOKE DATE=NONE   RESUME DATE=NONE
LAST-ACCESS=19.168/15:26:13
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED      (DAYS)          (TIME)
-----
ANYDAY              ANYTIME
GROUP=SYS1          AUTH=USE        CONNECT-OWNER=IBMUSER  CONNECT-DATE=19.167

```

```

CONNECTS=    02  UACC=NONE      LAST-CONNECT=19.168/15:26:13
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
  NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
Ready;

rac alu edi nospecial
Ready;

rac lu edi
USER=EDI NAME=UNKNOWN OWNER=IBMUSER  CREATED=19.167
DEFAULT-GROUP=SYS1      PASSDATE=19.167 PASS-INTERVAL= 30 PHRASEDATE=N/A
ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
LAST-ACCESS=19.168/15:26:13
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED      (DAYS)          (TIME)
-----
ANYDAY              ANYTIME
GROUP=SYS1          AUTH=USE        CONNECT-OWNER=IBMUSER  CONNECT-DATE=19.167
CONNECTS=    02  UACC=NONE      LAST-CONNECT=19.168/15:26:13
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
  NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
Ready;

```

For more information about RACF attributes, see *RACF Security Server Security Administrator's Guide*, SC24-6311.

5.1.2 RACF passwords and password phrases

Since RACF FL 530, password phrases can be defined for z/VM user IDs.

It is important to understand that *passwords* and *password phrases* are two different things. *Passwords* are uppercase by default or can be mixed case, if enabled by using the **RAC SETROPTS PASSWORD(MIXEDCASE)** command, and *password phrases* are mixed case by default. Passwords are 1 - 8 characters, and password phrases that use the default installation can be 14 - 100 characters. A user can be assigned a password, a password phrase, or both.

The default operation when creating a user profile is to not have a default value that is assigned to the password or the password phrase. The user is a protected user and cannot log on. This configuration is the preferable situation for disconnected service machines or guests user IDs.

For human IDs, the enterprise security policy defines what kind of authenticator is used. The initial password or password phrase of a user is not assigned by them.

When assigned a password or password phrase, the user can change that value at any time, but cannot remove it. When assigning the value for a user for the first time, make sure it is difficult to guess so the user has enough time to change it before someone else changes it. By default, the user ID is forced to change this initial value the first time it is used.

This section describes how to implement both functions.

Password and password phrases rules

Your organization's security policy is likely to have a section that describes the rules that govern system passwords. On z/VM with RACF installed, these rules are implemented with the **RACF SETROPTS** commands by a user with the SPECIAL attribute. The following parameters control password requirements:

- ▶ Password change interval
- ▶ Inactive virtual machine (VM) intervals
- ▶ When access is revoked because of unsuccessful login attempts
- ▶ Password history (password reuse)

For example, the following password and password phrases policies are available:

- ▶ **RAC SETROPTS PASSWORD(INTERVAL(90))** defines the change interval to 90 days.
- ▶ **RAC SETROPTS PASSWORD(MINCHANGE(5))** specifies that users cannot change their passwords more than once in 5 days, for example.
- ▶ **RAC SETROPTS INACTIVE(30)** revokes a user ID if it is unused for more than 30 days.
- ▶ **RAC SETROPTS PASSWORD(REVOKE(4))** defines the limit of successive incorrect use of passwords or password phrases before revoking the user.
- ▶ **RAC SETROPTS PASSWORD(HISTORY(6))** defines the number of previous passwords and password phrases that RACF saves for each user to avoid duplication.

Password syntax rules

Password syntax rules include (up to eight syntax rules) the following items:

- ▶ Password length
- ▶ Password character requirements (vowels, numbers, and so on)
- ▶ Password in mixed case

For example, the following password policies are available:

- ▶ **RAC SETROPTS PASSWORD(MIXEDCASE)** allows mixed-case passwords.
- ▶ **RAC SETROPTS PASSWORD(SPECIALCHARS)** allows special characters.

The following rules pertain to password verification and control to define the syntax of the new passwords for your installation:

- ▶ **RAC SETROPTS PASSWORD(RULE1(LENGTH(6:8) ALPHA(1) ALPHANUM(3:8)))** and **RAC SETROPTS PASSWORD(RULE2(LENGTH(8)))**
- ▶ **RAC SETROPTS PASSWORD(RULE1(LENGTH(8) VOWEL(1,3,5:8) NUMERIC(2,4)))** and **RAC SETROPTS PASSWORD(RULE2(LENGTH(8) MIXEDALL(1:8)))**

The **RACF SETROPTS LIST** command displays the password settings that are shown in Example 5-7.

Example 5-7 RACF SETROPTS LIST to display password settings

```
PASSWORD PROCESSING OPTIONS:
  PASSWORD CHANGE INTERVAL IS  90 DAYS.
  MIXED CASE PASSWORD SUPPORT IS IN EFFECT
```

6 GENERATIONS OF PREVIOUS PASSWORDS BEING MAINTAINED.
AFTER 4 CONSECUTIVE UNSUCCESSFUL PASSWORD ATTEMPTS,
A USERID WILL BE REVOKED.
NO PASSWORD EXPIRATION WARNING MESSAGES WILL BE ISSUED.
INSTALLATION PASSWORD SYNTAX RULES:

RULE 1 LENGTH(6:8) A*LLLLLL
RULE 2 LENGTH(8) *****

LEGEND:

A-ALPHA C-CONSONANT L-ALPHANUM N-NUMERIC V-VOWEL W-NOVOWEL *-ANYTHING
c-MIXED CONSONANT m-MIXED NUMERIC v-MIXED VOWEL \$-NATIONAL

Although the password verification is done by RACF when it is active, some user CP directory entries still have meaning, including the following examples:

- ▶ **NOLOG**: The user cannot log on to the system.
- ▶ **AUTOONLY**: The user can be only XAUTOLOGed. Having a user who is defined with NOPASSWORD and NOPHRASE attributes has the same effect.
- ▶ **NOPASS**: The user can log on without the use of a password. When the FACILITY class is not activated, or the IRR.NOPASS profile is not defined on the FACILITY class, any NOPASS user can log on without specifying a password. The security administrator should take extra care when the **NOPASS** statement is used.

Password phrases

A password phrase is a character string that consists of mixed-case letters, numbers, and any special characters, including blanks. Consisting of all those possibilities, password phrases have security advantage over passwords.

The **NOPASSWORD** attribute can be specified in **ADDUSER** or **ALTUSER** so that a user can authenticate only with a password phrase, which is stronger than a password.

Password phrases are implemented by default in RACF with a basic set of syntax rules. These syntax rules apply to all password phrases and cannot be altered or removed. However, the rules that install the ICHPWX11 exit can be enhanced. This section provides information about how to implement the exit.

RACF can use the new password phrase exit, ICHPWX11, to enhance RACF function when validating a new password phrase. This exit runs a REXX exec **IRRPHREX**. A sample is included by IBM in source form on the RACFVM 305 disk and consists of the exit, ICHPWX11, and a REXX exec named **IRRPHREX**.

ICHPWX11 must be installed as described in *RACF Security Server System Programmer's Guide*, SC24-6312. As included, all the checks are disabled and the exec is functionally equivalent to having no exit, but the following checks can be enabled in the REXX exec:

- ▶ Minimum length
- ▶ Maximum length
- ▶ List of allowable characters
- ▶ Leading blanks that are allowed or not
- ▶ Trailing blanks that are allowed or not
- ▶ Words in user name that is allowed or not
- ▶ Triviality checks
- ▶ Minimum unique characters by position from old password phrase
- ▶ Minimum unique words from old password phrase
- ▶ Dictionary check (hardcoded list of words)

The exit gains control when a new password phrase is processed. It can examine the value that is specified for the password phrase and enforce installation rules in addition to the RACF rules. For example, although RACF does not allow the user ID to be part of the password phrase, the exit can perform more complex tests such as disallow the company name, the names of months, and the current year in the password phrase.

The user of the new password phrase exit *augments* the RACF rules, but cannot override them. Be sure that the exit and the RACF rules do not contradict each other. For example, if the exit requires that the pass phrases contain all alphabetic characters, users cannot create password phrases because RACF requires at least two non-alphabetic characters. If you attempt to assign a phrase that conflicts the password rules, RACF does not accept the new phrase and displays the following message:

```
ICH21039I NEW PASS PHRASE REJECTED BY RACF RULES
```

The interval value that is specified on the **PASSWORD** command applies to passwords and password phrases. It continues to be processed by the new password exit, ICHPWX01, and is not passed to the ICHPWX11 exit.

For more information about implementing password phrases for RACF, see *RACF Security Server System Programmer's Guide, SC24-6312*. The HLASM product is required to assemble the ICHPWX11 file. If HLASM is not available, IBM provides a TEXT file that is assembled that can be used, as described in step 4 of the process.

Complete the following steps on the 7VMRAC10 VM:

1. Run **access 590 t**.
2. Run **vmfsetup 7vmrac10 racf**.
3. Run **copy ichpx11 assemble k = e**.
4. If HLASM is not available, complete the following steps:
 - a. Run **copy ichpx11 text k = txt00000 e**.
 - b. Go to step 11.
5. Remove the following comments by running **rpibllpa exec v**:

```
*:OBJNAME. ICHPWX11 LEPARMS RENT REUS LET NCAL XREF SIZE 100K,80K
*:OPTIONS. IGNORE
*:PARTID. ICHPWX11 TXT
*:EOBJNAME.
```
6. Run **vmfhlasm ichpx11 7vmrac10 racf (\$select outmode e**.
7. Run **rename ichpx11 txt00000 e = txt10001 e**.
8. Run **rename ichpx11 assemble e = asm10001 e**.
9. Run **vmfsim logmod 7VMRAC10 vvt1cl e tdata :mod 1cl0001 :part ichpx11 txt**.
10. Run **mfsim logmod 7VMRAC10 vvt1cl e tdata :mod 1cl0001 :part ichpx11 asm**.
11. Run **vmfbld ppf 7vmrac10 racf (serviced**.

Put the code into production (the copy files are created by VMFBLD to the RACFVM 305 disk).

Note: The process that is described in *RACF Security Server System Programmer's Guide*, SC24-6312, does not work as documented. When you link to the RACFVM 305 disk, you cannot get it in write mode because RACFVM has the disk in write mode. If you force off the RACFVM, you have no external security manager (ESM) and you cannot autolog RACMAINT after you forced RACFVM. This section describes how you can put the code into production.

For this process, you must give 7VMRAC10 the privilege class A or C so that it can run the **set secuser** command. You can use your normal processes to change the privilege class and then place the directory online. You must log off and then log on to the 7VMRAC10 VM to pick up the directory change. Then, run the **vmfsetup 7vmrac10 racf** command to reestablish your disk search order.

Perform the task that is shown in Example 5-8 to gain write access to the RACFVM 305 disk.

Example 5-8 Write access to the RACFVM 305 disk

```
set secuser racfvm 7vmrac10
HPCPFX6768I SECUSER of RACFVM initiated.
Ready; T=0.01/0.01 15:07:06
send cp racfvm det 305
Ready; T=0.01/0.01 15:07:14
RACFVM : DASD 0305 DETACHED
link racfvm 305 305 mr
RACFVM : (OPERATOR) ICH408I USER(7VMRAC10) GROUP(SYS1 ) NAME(#####
#####)
RACFVM : (OPERATOR) RACFVM.305 CL(VMMDISK )
RACFVM : (OPERATOR) INSUFFICIENT ACCESS AUTHORITY
RACFVM : (OPERATOR) ACCESS INTENT(CONTROL) ACCESS ALLOWED(NONE)
RPIMGR032E YOU ARE NOT AUTHORIZED TO LINK TO RACFVM.305
HCPLNM298E RACFVM 0305 not linked; request denied
Ready(00298); T=0.01/0.01 15:07:22
send cp racfvm link * 305 305 mr
RACFVM : DASD 0305 LINKED R/W
Ready; T=0.01/0.01 15:07:53
send racfvm acc 305 b/a
RACFVM : DMSACC724I 305 replaces B (305)
Ready; T=0.01/0.01 15:08:03
```

As shown in Example 5-8, a security violation with the **link** command. To solve it, use one of your systems RACF administrators and run the **racf permit** command to allow 6VMRAC30 to have *control* access to the RACFVM 305 disk:

```
rac permit racfvm.305 class(vmmdisk) id(7vmrac10) ac(control)
```

Now, you can complete the task of moving files to the RACFVM 305 disk, as shown in Example 5-9.

Example 5-9 Moving files to the RACFVM 305 disk

```
send racfvm det 305
RACFVM : DASD 0305 DETACHED
RACFVM : CST
Ready; T=0.01/0.01 15:17:45
link racfvm 305 305 mr
Ready; T=0.01/0.01 15:17:55
acc 305 z
Ready; T=0.01/0.01 15:18:01
```

```

vmfcopy * * k = = z (prodid 7vmrac10%racf oldd replace
Ready; T=0.25/0.33 15:19:35
det 305
DASD 0305 DETACHED
Ready; T=0.01/0.01 15:19:46
send racfvm link * 305 305 mr
RACFVM : CST
Ready; T=0.01/0.01 15:19:57
send racfvm access 305 b/a
RACFVM : DMSACC724I 305 replaces B (305)
Ready; T=0.01/0.01 15:20:07
RACFVM : DMSACP723I B (305) R/O
RACFVM : CST

```

Then, run the RACFVM **ipl 490** command that restarts RACF, as shown in Example 5-10. You cannot perform an IPL of CMS or 190 in RACFVM, or RACF does not start correctly.

Example 5-10 RACF performs an IPL of 490

```

send cp racfvm ipl 490 clear parm autocr
Ready;
RACFVM : RACFVM CMS XA Rel. 27 2011-10-18
RACFVM : DMSACP723I B (305) R/O
RACFVM : DMSACP723I T (190) R/O
RACFVM : RACF is defined to the Z/VM system and the current product status is E
NABLED
RACFVM :
RACFVM : RACF
RACFVM : Feature for z/VM
RACFVM : Version 7.1.0
RACFVM :
RACFVM : Licensed Materials - Property of IBM
RACFVM : 5741-A07
RACFVM : (C) Copyright IBM CORP. 1981, 2012 All Rights Reserved.
RACFVM :
RACFVM : DMSACC723I R (0200) R/W - OS
RACFVM : DMSACC723I Q (0300) R/W - OS
RACFVM : CSTSET001I CMS SUB-TASKING SUPERVISOR INITIALIZED.
RACFVM : CSTINT003I INITIATOR ACTIVATED.
RACFVM : ICH508I ACTIVE RACF EXITS: ICHRCX02
RACFVM : ICH520I RACF 7.1.0 IS ACTIVE.
RACFVM : RPISTR001I Program CSTDYNST Initiated.
15:37:15 * WNG FROM RACFVM : RACF/VM SERVICES ARE NOW AVAILABLE.
RACFVM : * WNG FROM RACFVM : RACF/VM SERVICES ARE NOW AVAILABLE.
RACFVM : RPISTR002I Program CSTDYNST Ended. Completion code = 000000.
RACFVM : RPISTR003I Subtask RPIMSG Initiated.
RACFVM : RPISTR003I Subtask RPIINIT Initiated.
RACFVM : RPICLS104W - DEFAULT SETTINGS WERE MADE FOR ALL AUDITABLE AND
RACFVM : CONTROLLABLE VM EVENTS.
RACFVM : RPICLS123I RACF Extended password support registered with CP
RACFVM : RPIMGR003I 15:37:15: CONNECTION COMPLETE TO CP ON PATHID 0000
RACFVM : RACF AUTHORIZATION COMMUNICATION INTERFACE READY
set secuser racfvm reset
RACFVM : HCPCFX6768I Your SECUSER set to RACFVM by EDI.
HCPCFX6769I SECUSER of RACFVM terminated.
Ready; T=0.01/0.01 15:38:03

```

Note: This process was the only way that you can allow the RACFVM 305 disk to be updated without a system outage. If you can accept the outage, you shut down the system and perform an IPL with the **NOAUTOLOG** parameter. Then, start RACMAINT as described.

At this point, the sample exit does not perform any other functions compared to having no exit. You now adjust the exit to reflect your installation requirements.

Password phrase syntax rules

Password phrase syntax must adhere to the following rules:

- ▶ Maximum length: 100 characters
- ▶ Minimum length:
 - Nine characters when ICHPWX11 is present and allows the new value.
 - A total of 14 characters when ICHPWX11 is not present.
- ▶ Must not contain the user ID (as sequential uppercase or sequential lowercase characters).
- ▶ Must contain at least two alphabetic characters (A - Z, a - z).
- ▶ Must contain at least two non-alphabetic characters (numerics, punctuation, or special characters).
- ▶ Must not contain more than two consecutive characters that are identical.
- ▶ Must be enclosed in single quotation marks, with single quotation marks within the password phrase doubled. The quotation marks must be removed from the password phrases when RACF prompts at logon.
- ▶ Must not contain forward slashes, nulls (X'00'), or leading or trailing blanks.

Only a RACF administrator can assign the initial phrase. When assigned, the user can modify the phrase, and is prompted to change it by default the first time it is used to log on.

To disable the password function and enable a phrase, run the following command:

```
rac alu edi nopassword phrase('it is friday')
```

When the VM EDI logs on to the system, it is prompted to change the password. When changing the password from the logon prompt, do not use the quotation marks (for example, 'red white blue' should be red white blue).

If the VM wants to change the phrase while logged on to the system, run the following command:

```
rac phrase phrase('red white blue' 'howdy to everyone in vm land')
```

Although it looks like a mistake, the command is correct. It is **phrase** and it includes an operand of **phrase**.

It is possible to adjust the z/VM logo to accept more than eight characters in the password field, so the use of the command line is not needed for password phrases. IBM provides a utility program that is called **DRAWLOGO** and a sample XEDIT macro called X\$DRWL\$X at CP sample disk, 2C2, on the MAINT710 user. To use the utility, rename **DRAWLOGO** **SAMPEXEC** to **DRAWLOGO EXEC** and X\$DRWL\$X **SAMPXEDI** to X\$DRWL\$X **XEDIT**.

Open the input file (the default is INPTAREA SAMPLE on PMAINT CF0 disk) with the **DRAWLOGO** utility:

```
drawlogo INPTAREA SAMPLE B
```

Press PF5 and use the Settings menu to select the length of user ID and password input area. Place the cursor in the position you want the password input field to start and use PF4 to access the Input menu. Pressing PF4 again fills the password input area with the characters for password input. Press PF11 to display the results.

RACF user passwords encryption

RACF provides the following algorithms for authenticating passwords and password phrases:

- ▶ Masking
- ▶ Data Encryption Standard (DES) algorithm
- ▶ Key Derivation Function with AES256 (KDFAES) algorithm for passwords

The masking algorithm is the original algorithm that is provided with RACF. The RACF DES algorithm provides a higher level of security than the masking algorithm and is identified in the Federal Information Processing Standard 46-1 of the Computer Systems Laboratory in Gaithersburg, Maryland, of the National Institute of Standards and Technology of the United States Government.

DES is accepted as a national and international standard. The KDFAES algorithm provides the highest level of security, and is resistant to offline attacks. When installing RACF on your system, the DES algorithm is the default algorithm.

RACF also supports an installation-defined method that is implemented that uses the ICHDEX01 exit. For more information about ICHDEX01, see “RACF Installation Exits”, in *RACF Security Server System Programmer's Guide*, SC24-6312.

To display the current enabled algorithms, use the **RACF SETROPTS LIST** command. Example 5-11 show an excerpt from the command output.

Example 5-11 Password excerpt from RACF SETROPTS LIST command

```
PASSWORD PROCESSING OPTIONS:
  THE ACTIVE PASSWORD ENCRYPTION ALGORITHM IS LEGACY
  PASSWORD CHANGE INTERVAL IS  30 DAYS.
  PASSWORD MINIMUM CHANGE INTERVAL IS  0 DAYS.
  MIXED CASE PASSWORD SUPPORT IS NOT IN EFFECT
  SPECIAL CHARACTERS ARE NOT ALLOWED.
  NO PASSWORD HISTORY BEING MAINTAINED.
  USERIDS NOT BEING AUTOMATICALLY REVOKED.
  NO PASSWORD EXPIRATION WARNING MESSAGES WILL BE ISSUED.
  NO INSTALLATION PASSWORD SYNTAX RULES ARE PRESENT.
```

In general, encryption programs are a two-way process: encryption and decryption.

Encryption processes use the data and an encryption key to create the encrypted form of the data.

Decryption is the reverse operation, and uses the encryption key and the encrypted form of the data to re-create the original data.

When configured to do so, RACF uses the encryption algorithms to encrypt the password and store it on the database. Because RACF does not store the password that is used as the encryption key, until now, the original data cannot be reconstructed. That is, meaning that the password that is encrypted and stored in the RACF database cannot be decrypted. With this one-way process, RACF provides a high level of security.

This fact does not mean that any user on the system can have READ access to the RACF database. Use the Least Privilege Principle and give READ access only to the users that really need it for their jobs.

By default, RACF uses the password or password phrase as an encryption key to encrypt the user ID and store it in the RACF database by using the DES algorithms to authenticate a user on the system. When a user must log in, RACF again encrypts the user ID by using the password or password phrase that is provided during the login and compares it with the encrypted data in the RACF database. If the data matches, the password or passphrase is valid.

RACF KDFAES algorithm

The KDFAES algorithm is one of the available encryption algorithms in RACF that is used to encrypt password and password phrases. It requires enablement of CPACF, which is a no-charge feature on your hardware (FC 3863). This algorithm is preferred among others that are available because it is more secure to offline attacks because it incorporates the following properties:

- ▶ Each instance of a RACF password uses randomly generated text in the encryption process, which prevents the use of pre-computed password hashes. An offline attack must perform the full encryption process for every password guess, as opposed to comparing the password hash against a list of pre-computed values. This configuration slows down the attack, which makes it take much longer to guess passwords.
- ▶ Thousands of hash operations are performed against the password and random text to generate a key. That key is then used to encrypt the user ID. An offline attack also is slowed, which must perform the same number of operations for each password guess. However, the authorized user logging on to the system that uses their clear text password does not notice the increased processing effect.

To enable the KDFAES algorithm for password and password phrases, run the SETROPTS command, as shown in Example 5-12.

Example 5-12 Enabling the KDFAES encryption algorithm

```
rac setropts password(algorithm(kdfaes))
Ready;

rac setropts list
...
PASSWORD PROCESSING OPTIONS:
  THE ACTIVE PASSWORD ENCRYPTION ALGORITHM IS KDFAES
  PASSWORD CHANGE INTERVAL IS 30 DAYS.
  PASSWORD MINIMUM CHANGE INTERVAL IS 0 DAYS.
  MIXED CASE PASSWORD SUPPORT IS NOT IN EFFECT
  SPECIAL CHARACTERS ARE NOT ALLOWED.
  NO PASSWORD HISTORY BEING MAINTAINED.
  USERIDS NOT BEING AUTOMATICALLY REVOKED.
  NO PASSWORD EXPIRATION WARNING MESSAGES WILL BE ISSUED.
  NO INSTALLATION PASSWORD SYNTAX RULES ARE PRESENT.
...
Ready;
```

Make sure that you review “Planning Considerations for Enabling KDFAES” in *RACF Security Server System Programmer’s Guide*, SC24-6312 before you enable KDFAES.

After enabling the KDFAES algorithms, passwords that are encrypted with the DES algorithm continue to be evaluated properly by RACF. User passwords do not need to be changed.

When the users change their passwords, the process is encrypted by using the KDFAES algorithm. The **PWCONVERT** operand of the **ALTUSER** command can be used to transform a password that is encrypted with the DES algorithm (but not a password phrase) into a password that is encrypted with KDFAES without requiring the password to be changed.

If you have backups of the RACF database containing passwords that were encrypted by using DES or masking, they are more susceptible to offline attacks. If the hash represents the same clear text password as the user's current password, and an attacker can guess the value, it can be used to log on to the user's account, even if the current password is encrypted by using KDFAES.

The **EXPIRED** operand of the **ALTUSER** command can be used to mark a password as expired, which requires it to be changed at the next logon. This operand can help accelerate the password change process.

If previous passwords were encoded by using the masking algorithm, they must be changed. They are not be properly evaluated when KDFAES is enabled, and cannot be converted to KDFAES by using the **PWCONVERT** function.

5.1.3 Implementing RACF LOGONBY

RACF supports the LOGONBY function with the SURROGAT class facility, but is not limited to the maximum of eight surrogate VMs. The RACF LOGON BY acts the same way as the CP LOGONBY function, which allows authorized VMs to log on to a shared VM by using their own password.

To implement the RACF LOGON BY facility, complete the following steps:

1. Run the **setropts** command to activate the CLASSACT(SURROGAT) class:

```
rac setropts class(surrogat)
```

2. Verify that the SURROGAT class is active:

```
rac setr list
```

Example 5-13 shows the output of the command.

Example 5-13 Enabling the SURROGAT class

```
rac setropts class(surrogat)
Ready;
  rac setr list
ATTRIBUTES = INITSTATS NOWHEN(PROGRAM)
STATISTICS = NONE
ACTIVE CLASSES = DATASET USER GROUP VMMDISK VMRDR VMBATCH VMLAN VMSEGMT
                  FACILITY SURROGAT VXMBR VMXEVENT XFACILIT GXFACILI
```

3. Define the profiles of the form LOGONBY.shared_userid in the SURROGAT class for each user ID that is shared.
4. Permit specific users for the appropriate SURROGAT profiles.
5. List the information by running the **RLIST** command.

LOGON BY processing

As a preferred practice, create a sample file from which to copy to implement the LOGON BY function, as shown in Example 5-14, where you change *shrduser* and *surrogat-id1*.

Example 5-14 RPIDIRECT SURROGAT

```
RPIDIRECT SURROGAT A1 F 80 Trunc=80 Size=5 Line=0 Col=1 Alt=0
====>
0 * * * Top of File * * *
1 RDEFINE SURROGAT LOGONBY.shrdusr UACC(NONE) AUDIT(ALL)
2 PERMIT LOGONBY.shrdusr CL(SURROGAT) RESET(ALL)
3 PERMIT LOGONBY.shrdusr CL(SURROGAT) ACC(READ) ID(surrogat-id1)
4 PERMIT LOGONBY.shrdusr CL(SURROGAT) ACC(READ) ID(surrogat-id2)
5 RL SURROGAT LOGONBY.shrdusr AUTH
6 * * * End of File * * *
```

When you must add surrogate users to the RACF database, copy this file to RPIDIRECT SYSUT1 on your A disk and then modify that file, as shown in Example 5-15.

Example 5-15 RPIDIRECT SYSUT1 before the changes

```
RPIDIRECT SYSUT1 A1 F 80 Trunc=80 Size=5 Line=0 Col=1 Alt=0
====> ch /shrdusr/MAINT/* *
0 * * * Top of File * * *
1 RDEFINE SURROGAT LOGONBY.shrdusr UACC(NONE) AUDIT(ALL)
2 2 PERMIT LOGONBY.shrdusr CL(SURROGAT) ACC(ALTER) ID(shrdusr) RESET(ALL)
3 PERMIT LOGONBY.shrdusr CL(SURROGAT) ACC(READ) ID(surg-id1)
4 PERMIT LOGONBY.shrdusr CL(SURROGAT) ACC(READ) ID(surg-id2)
5 RL SURROGAT LOGONBY.shrdusr AUTH
6 * * * End of File * * *
```

If you want to add SURROGAT support for the MAINT VM, tailor the file to look like Example 5-16.

Example 5-16 RPIDIRECT SYSUT1 after the changes

```
RPIDIRECT SYSUT1 A1 F 80 Trunc=80 Size=8 Line=0 Col=1 Alt=0
====>
0 * * * Top of File * * *
1 ALTUSER MAINT NOPASSWORD NOPHRASE
2 RDEFINE SURROGAT LOGONBY.MAINT UACC(NONE) AUDIT(ALL)
3 PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(ALTER) ID(MAINT) RESET(ALL)
4 PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(PWNOVAK)
5 PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(PACOSTA)
6 PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(BADER)
7 PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(EDI)
8 PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(VIC)
9 PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(KLAUSM)
10 PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(WILLIANR)
11 RL SURROGAT LOGONBY.MAINT AUTH
12 * * * End of File * * *
```

ALTUSER MAINT NOPASSWORD NOPHRASE is a good way to protect the MAINT user ID from being revoked because of too many attempts with the wrong password. Before z/VM 5.3, MAINT was revoked by logging on directly with too many incorrect passwords. Since the 5.3 release, the ID is protected from this type of attack if you set **MAINT NOPASSWORD**.

In our example, we show the **PERMIT** for each user, although defining the permission by group (for example, ITSOGRP) is a preferred practice. Run **RPIBLDDS EXEC** again to run these definitions, as shown in Example 5-17.

Example 5-17 Output of RPIBLDDS

```

rpibldds rpidirct sysut1
Processing batch file RPIDIRCT SYSUT1 using "RAC" command interface
=> RDEFINE SURROGAT LOGONBY.MAINT UACC(NONE) AUDIT(ALL)
=> PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(ALTER) ID(MAINT) RESET(ALL)
=> PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(PWNOVAK)
=> PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(PACOSTA)
=> PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(BADER)
=> PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(EDI)
=> PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(VIC)
=> PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(KLAUSM)
=> PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(WILLIANR)
=> RL SURROGAT LOGONBY.MAINT AUTH
CLASS      NAME
-----
SURROGAT   LOGONBY.MAINT
LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----
00     IBMUSER      NONE                READ         NO
INSTALLATION DATA
-----
NONE
APPLICATION DATA
USER      ACCESS    ACCESS COUNT
-----
IBMUSER   ALTER      000000
PWNOVAK   READ       000000
PACOSTA   READ       000000
BADER     READ       000000
EDI       READ       000000
VIC       READ       000000
KLAUSM    READ       000000
WILLIANR  READ       000000

```

To use the LOGON BY function, log on with the **BY** keyword, as shown in Example 5-18.

Example 5-18 Log on by using the BY option

```

z/VM ONLINE      Welcome to the IBM z/VM Enterprise Virtualization Platform
ESM: RACF/VM      / VV      VVV MM      MM
                  / VV      VVV MMM     MMM
                  ZZZZZZ / VV      VVV  MMMM   MMMM
                   ZZ    / VV      VVV   MM MM MM MM
                   ZZ    / VV  VVV   MM  MMM  MM
                   ZZ    / VVVVV   MM   M   MM
                   ZZ    / VVV     MM     MM
                  ZZZZZZ / V      MM      MM
built on IBM Virtualization Technology    www.ibm.com/vm

  I T S O : ( S ) : ( ) International Technical
                               Support Organization
                               www.ibm.com/redbooks

```

Fill in your USERID and PASSWORD and press ENTER

(Your password will not appear when you type it)

USERID ==>

PASSWORD ==>

COMMAND ==> **1 maint by edi**

RUNNING ITS0ZVM4

When you are prompted for the password, the password for the VM EDI is supplied (see Example 5-19), although the VM MAINT is logged on.

Example 5-19 Logon complete

1 maint by edi

Enter your password,

or

To change your password, enter: ccc/nnn/nnn

where ccc = current password, and nnn = new password

```

ICH70001I MAINT      LAST ACCESS AT 11:20:40 ON TUESDAY, JUNE 11, 2019
HCPLNM102E DASD 0123 forced R/O; R/W by DIRMSAT4
z/VM Version 7 Release 1.0, Service Level 1801 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES: 0023 RDR,   NO PRT, 0002 PUN
LOGON AT 16:27:39 EDT TUESDAY 06/11/19
z/VM V7.1.0      2019-05-18 16:18

```

For more information, see *RACF Security Server System Programmers's Guide*, SC24-6312, and *RACF Security Server Administrator's Guide*, SC24-6311.

5.2 Communication encryption

Correctly implementing and managing security controls for the z/VM hypervisor is a mandatory cornerstone, no matter how large or small your enterprise is. Your security posture is only as strong as the weakest point, which means that the correct encryption of traffic must be implemented at all layers.

Connectivity to the hypervisor layer and well-secured guests on an unsecured hypervisor are critical exposures. Furthermore, in nearly all circumstances, encrypting traffic as a default practice is common sense. Encryption requirements might also be mandated by company policy, clients, partners, vendors, industry regulations, or governing bodies.

The use of encrypted communication can increase the security of the IT infrastructure and should always be listed in the company security policy. By default, Telnet 3270 session data flows decrypted over the network, in clear text, meaning that anyone who dumps the network traffic can see what is happening between the Telnet client and the z/VM 3270 connection.

Transport Layer Security (TLS), and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols to provide end-to-end encrypted communication. Digital certificates and trust hierarchies can be implemented to use encrypted communication. Dynamic SSL/TLS connections are supported by the following z/VM TCP/IP application servers and clients, which are updated to accommodate this support:

- ▶ TCP/IP server
- ▶ SSL server
- ▶ FTP server
- ▶ FTP client
- ▶ Telnet server (internal to the TCP/IP server)
- ▶ Telnet client
- ▶ Simple Mail Transfer Protocol (SMTP) server

When discussing SSL/TLS for z/VM, SSL* is a pool of CMS VMs that provide encrypted communication to clients that are connecting to z/VM. Its code is preinstalled as part of a standard z/VM installation and can be customized and enabled to provide SSL/TLS connections.

Most of the TCP/IP stack service machines can have security that is controlled by RACF. This configuration allows RACF to process user ID authentication and authorization to the system and to resources, which increases the level of security on the system.

For more information about how to customize and enable encrypted communications to and from z/VM, see Chapter 4, “Installing and configuring z/VM”, in *The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3*, SG24-8147.

5.3 Single System Image Security

This section covers the security implications of using z/VM 7.1 with Single System Image (SSI) and live guest relocation (LGR).

This z/VM feature was introduced with z/VM 6.2 and it provides more flexibility for your environment by allowing Linux guests to be moved from one logical partition (LPAR) to another. It also provides a set of shared resources for member systems and their Linux guests.

Rather than managing security of a single implementation on a single device, administrators can manage the security of two or more operating systems. This control of access is a useful mechanism for helping to protect your data. In your SSI cluster, use storage area network (SAN) zoning and logical unit number (LUN) masking to ensure that data is available only for servers that should access it.

This section also explains the concept of relocating domains to control the relocation of Linux on IBM LinuxONE guests to provide flexibility and availability to meet user demands for security and manageability.

5.3.1 Overview

VMs allow quick turnaround and flexibility for multiple projects and environments. To benefit from VMs, z/VM uses virtualization and gives administrators the power to manage their resources on the IBM LinuxONE platform.

With the IBM z/VM SSI, which was introduced with z/VM v6.2, a running Linux on IBM LinuxONE VM can be relocated from one member system to any other, a process known as *LGR*. Support for LGR allows you to move Linux virtual servers without disruption to the business, which helps you to avoid planned outages. The z/VM systems are aware of each other and can benefit from their combined resources.

LGR enables clients to avoid loss of service because of planned outages by relocating guests from a system requiring maintenance to a system that remains active during the maintenance period. This capability can be used to move workloads from one z/VM LPAR to another when needed. It also helps to reduce planned outages during hardware changes or a z/VM initial program load (IPL). However, the environment becomes more complex and requires special attention with the shared resources. With this feature in place, it is important to know how to manage efficiently the configuration in a secure manner.

LGR brings more flexibility to your environment, not just Linux availability. Now, you can use this new feature to build a reliable and secure infrastructure for Linux guests running under z/VM by workload balancing. Also, hardware and z/VM changes can be run without affecting service availability.

Note: SSI is included in the base of z/VM V7.1 at no extra cost. Previously, it was a priced feature of z/VM V6, and is withdrawn. Integrating and making SSI available at no charge is intended to help more clients reduce or shorten planned outages of their Linux workloads as they adopt the z/VM Continuous Delivery model for their z/VM systems.

5.3.2 Equivalency identifiers

With an SSI cluster, you use EQIDs. EQIDs are used to ensure that all members of the cluster use both the same physical devices and the devices that are attached over IBM Fibre Channel Connection (IBM FICON). During z/VM IPL, the EQID number is automatically generated and assigned to various devices, such as DASDs. However, for Fibre Channel Protocol (FCP) devices, you must explicitly set the EQID on the system config file on the PMAINT CF0 disk so that all cluster members can see the device as one device.

Before a Linux guest is relocated to a different z/VM LPAR, the guest configuration is checked on the destination LPAR (Device condition) to ensure that the devices have the same EQIDs. Ensure that all necessary FCP configuration is planned to avoid problems when relocating your Linux guests. To update the NPIV devices, review and if necessary update your EQIDs as well.

For the list of conditions, see *z/VM: CP Planning and Administration*, SC24-6271.

5.3.3 Relocation domains

A relocation domain defines a set of members of an SSI cluster among which VMs can relocate freely. A domain can be used to define the subset of members of an SSI cluster to which a particular guest can be relocated. Relocation domains can be defined for business or technical reasons.

For example, a domain can be defined that has all of the architectural facilities necessary for a particular application, or a domain can be defined to allow access only to systems with a particular software tool. Whatever the reason for the definition of a domain, CP allows relocation among the members of the domain without any change to architectural characteristics or CP functions as seen by the guest.

The relocation domain for a VM can be defined with the **VMRELOCATE** directory statement. When the user ID logs on, CP assigns a virtual architecture level to the VM that is the maximal common subset of the architectural features (hardware architecture facilities and CP-supplied features) of all the members of the SSI cluster that belong to that relocation domain. The guest cannot use architectural features that are not included in this virtual architecture level. This ensures that the guest can be freely relocated to other members of the domain because they provide the same architectural features. A feature must be supported in every relevant component (processor, channel, and device hardware, and the z/VM software level) on every domain member to be usable to the guest.

The following types of relocation domains are defined implicitly:

- ▶ A domain that includes all of the members of the SSI cluster. The name of this domain is SSI.
- ▶ A domain that includes one member of the SSI cluster. A single-member domain is defined for each member. The name of the domain is the member's system name.

When a user ID that is defined by a single-configuration VM definition logs on, the default associated relocation domain is the entire SSI cluster (domain SSI), unless a different relocation domain is set by a **VMRELOCATE** statement in the user's VM definition.

If relocation of the VM is disabled on the **VMRELOCATE** statement and no relocation domain is specified, the default relocation domain is the single-member domain of the system where the user logs on, and the user is assigned a virtual architecture level that is the set of all the architectural features of that system.

Note: When a Linux guest is allowed to relocate to only a subset of the members in a cluster with the respective relocation domain, it can be overruled by using the force option of the **vmrelocate** command.

5.3.4 RACF in an SSI cluster

When RACF is installed in a z/VM SSI environment, it is mandatory that the RACF database is shared. To ensure database integrity, the following requirements must be met:

- ▶ The RACF database DASD is defined as shared in the I/O configuration.
- ▶ The primary RACF database (device 200) and the backup database (device 300) are defined on full-pack minidisks.
- ▶ These devices include virtual reserve/release as enabled.

Use the **DEVNO** operand of the **MDISK** directory statement to define the DASD as full-pack minidisks.

If you are following the preferred practice of using the same real device numbers across LPARs to reference DASD, the **MDISK** statements for the RACF database disks can be placed in the identity entry for the RACF server. If the real device numbers are not the same across LPARs, the **MDISK** statements must be placed in the relevant subconfiguration entries.

5.4 Auditing

A defined information security policy is worthless if you cannot assess whether the policies are effective, meaning that it was adhered to by all employees and they are playing the roles of which they are expected.

Tracking changes and authorized and unauthorized accesses is a way to make sure that the information security policy is followed. But again, with the increase of servers that are managed on the IT infrastructure, the amount of audit data that is generated makes it impossible for a human to analyze all of it, find a threat, and act on it while the intrusion is still occurring. For that reason, define during the planning stage of the IT infrastructure which actions must be logged for audits.

The complexity in auditing is reduced when defined roles are available in the information security policy. Users under one role should not have access to override the mandatory access controls (MACs) and should not manipulate the controls that are under the jurisdiction of another job role. With the separation of duties, the functions of the systems and integrity of audit logs are not compromised.

In z/VM, audit trails are generated by several CP command journaling options. They can be used to identify unsuccessful attempts of a CP command use. When journaling is turned on, more information is recorded with unsuccessful and successful attempts of specific CP commands.

For a comprehensive audit trail, the use of an ESM is recommended. In this book, we cover the auditing with the use of RACF/VM. It can audit every command and security-relevant event happening within the hypervisor, in accordance with a predefined security policy.

5.4.1 Auditing with journaling

z/VM offers a mechanism to track unauthorized **LOGON** attempts and unauthorized **LINK** commands. By enabling journaling, it is possible to configure how the system records **LOGON** and **LINK** attempts. Although it is fine for exploring **LOGON** attempts and unauthorized **LINK** commands, it really is not sufficient for the modern enterprise.

Enabling journaling

To use z/VM journaling, you must enable it in **SYSTEM CONFIG** and the system must have an IPL performed with the new configuration.

Example 5-20 shows an excerpt from a **SYSTEM CONFIG** file that is used to enable journaling (line numbers are not part of the **SYSTEM CONFIG** and are used to explain the statements on the lines).

Example 5-20 Configure journaling in SYSTEM CONFIG

```
1. Journaling,
2.     Facility          on,
3.     Set_and_Query    on,
4. Logon,
5. Message after 3 attempts to willianr,
6.     Account after 5 attempts,
7.     VM_Logo after 7 attempts,
8.     Lockout after 9 attempts for 10,
9. Link,
10. Message after 3 attempts to willianr,
11.     Account after 4 attempts,
12.     Disable after 5 attempts
```

Example 5-20 features the following lines:

- ▶ Line number 1 starts the journaling configuration statement in the **SYSTEM CONFIG** file.
- ▶ Line number 2 enables or disables journaling when the system undergoes an IPL.
- ▶ Line number 3 enables or disables the ability to set and query journaling. When disabled, the only configuration that takes effect is the configuration in the **SYSTEM CONFIG** file. It is not possible to use **query journaling** or **set journaling** commands, as shown in Example 5-21.

Example 5-21 Query and set journaling when Set_and_Query is off

```
q journal
HCPJRL003E Invalid option - JOURNAL
Ready(00003);

set journal link off
HCPJRL003E Invalid option - JOURNAL
Ready(00003);

set journal logon off
HCPJRL003E Invalid option - JOURNAL
Ready(00003);
```

- The journaling statement of z/VM allows the CP to control two kinds of actions: **LOGON** and **LINK**. The **LOGON** parameter starts on line 4 of Example 5-20 on page 131 and the **LINK** parameter starts on line 9 of the same example. Although the parameter is called **LOGON**, it also tracks successive tentatives of AUTOLOG and XAUTOLOG with an incorrect password in addition to the **LOGON** command.

For both parameters, it is possible to configure two options: **MESSAGE** and **ACCOUNT**. The **MESSAGE** parameter sets up the number of possible tries before a user receives an information message. Although any user can be set to receive the information message, setting a user that has the console logged is preferred as it is possible to look for the information later after the event happened.

In Example 5-22, user EDI tries to log on repeatedly with an incorrect password. The **MESSAGE** parameter was set to user BOB.

Example 5-22 Repeated logon attempts with incorrect password

#Logon tried repeatedly, but just one output is shown:

```
1 edi
ENTER PASSWORD (IT WILL NOT APPEAR WHEN TYPED):
```

```
HCPLGA050E LOGON unsuccessful--incorrect password
```

Enter one of the following commands:

LOGON userid	(Example: LOGON VMUSER1)
DIAL userid	(Example: DIAL VMUSER2)
MSG userid message	(Example: MSG VMUSER2 GOOD MORNING)
LOGOFF	

#After third try, user bob receives the information message:

```
HCPJRL145I User EDI at 9.12.5.143 issued a LOGON command with an invalid password
003 times. The limit is 003.
```

The same happens with successive **LINK** command attempts with an incorrect password to access a protected minidisk. Example 5-23 shows user EDI trying to link to the 191 protected minidisk of user BOB with an incorrect password.

Example 5-23 User EDI try to link a protected minidisk

```
link bob 191 191 rr
ENTER READ PASSWORD:
```

```
HCPLNM114E BOB 0191 not linked; mode or password incorrect
Ready(00114); T=0.01/0.01 16:30:21
```

```
link bob 191 191 rr
ENTER READ PASSWORD:
```

```
HCPLNM114E BOB 0191 not linked; mode or password incorrect
Ready(00114); T=0.01/0.01 16:30:30
```

```
link bob 191 191 rr
ENTER READ PASSWORD:
```

```
HCPLNM114E BOB 0191 not linked; mode or password incorrect
Ready(00114); T=0.01/0.01 16:30:35
```

#After third try, user bob receives the information message:

HCPJRL145I User EDI at 9.12.5.143 issued a LINK command with an invalid password 003 times. The limit is 003.

#All successive try will generate an information message:

HCPJRL145I User EDI at 9.12.5.143 issued a LINK command with an invalid password 004 times. The limit is 003.

HCPJRL145I User EDI at 9.12.5.143 issued a LINK command with an invalid password 005 times. The limit is 003.

The **ACCOUNT** parameter sets up the number of possible tries before CP detects that a user entered enough **LINK** commands to a protected minidisk that is not owned by the user with an invalid password that reaches or exceeds an installation-defined threshold value recording a type 06 accounting record and a type 04 accounting record. Then, CP detects that a user entered enough **LOGON**, **AUTOLOG**, or **XAUTOLOG** invocations with an invalid password that reaches or exceeds an installation-defined threshold value. A type 05 accounting record is generated when CP detects that a user successfully entered a **LINK** command to a protected minidisk that is not owned by the user.

In Example 5-24, user EDI continues repeatedly to try to log on with an incorrect password. The accounting record is created after the fifth try.

Example 5-24 Repeatedly try to log on and generate type 04 accounting records

#Logon tried repeatedly, but just one output is shown:

1 edi

ENTER PASSWORD (IT WILL NOT APPEAR WHEN TYPED):

HCPJRL145I User EDI at 9.12.5.143 issued a LOGON command with an invalid password 004 times. The limit is 003.

Enter one of the following commands:

LOGON userid	(Example: LOGON VMUSER1)
DIAL userid	(Example: DIAL VMUSER2)
MSG userid message	(Example: MSG VMUSER2 GOOD MORNING)
LOGOFF	

#All the successive logon attempts generate an information message:

HCPJRL145I User EDI at 9.12.5.143 issued a LOGON command with an invalid password 004 times. The limit is 003.

HCPJRL145I User EDI at 9.12.5.143 issued a LOGON command with an invalid password 005 times. The limit is 003.

HCPJRL145I User EDI at 9.12.5.143 issued a LOGON command with an invalid password 006 times. The limit is 003.

#After the fifth try, an accounting record is created:

EDI 060916170056L00412	0505	TCPIP	090C058F04
EDI 060916170118L004TESTE	0605	TCPIP	090C058F04

In Example 5-25, user EDI continues repeatedly to try to link to the 191 protected minidisk of user BOB with an incorrect password.

Example 5-25 Repeatedly try to link to a protected minidisk and generate type 06 accounting records

link bob 191 191 rr

ENTER READ PASSWORD:

HCPLNM114E BOB 0191 not linked; mode or password incorrect
Ready(00114); T=0.01/0.01 16:30:35

#After the fourth try, an accounting record is created:

EDI0060916163447L006PASSBOB 0404 0191 TCPIP 090C058F06

For more information about the accounting record format, see Chapter 7, “Setting Up Service Virtual Machines”, in *Accounting Record Formats z/VM V7.1 CP Planning and Administration*, SC24-6271.

The **LOGON** statement includes two more parameters: **VM_Logo** and **Lockout**. When **VM_Logo** is set, after the number of attempts that are specified by using the wrong password to log on, the user is redirected back to the z/VM Logo panel. When the number of attempts by using the wrong password reaches the **LOCKOUT** number, this user ID cannot be logged on for the number of minutes specified on the **LOCKOUT** parameter.

In our example, after nine uses of the wrong password, the user ID cannot be logged on for 10 minutes. The accounting record is still recorded, the information message is sent to the listed user, and the user trying to log on receives a message stating the maximum number of attempts were exceeded, as shown in Example 5-26.

Example 5-26 User who is locked out after excessive logon attempts

l edi

HCPJRL780E Maximum password attempts exceeded, try again later.

Accounting records:

EDI 060916170126L004123	0705	TCPIP	090C058F04
EDI 060916170142L005	0805	TCPIP	090C058F04
EDI 060916170148L005123	0905	TCPIP	090C058F04

Analogous to the **LOCKOUT** parameter for **LOGON**, it is possible to set the parameter **DISABLE** for **LINK** to disable the **link** command for the user that reached the maximum number of incorrect passwords while trying to link a protected minidisk. Accounting information is still recorded and information message sent to the user ID listed on the Message parameter. The user that is disabled from running a **LINK** command receive a message like the one shown in Example 5-27.

Example 5-27 User who is disabled from running LINK command after excessive link attempts

link bob 191 191 rr

HCPLNM115E LINK invalid; excessive incorrect passwords
Ready(00115); T=0.01/0.01 16:35:04

Accounting records:

EDI060916163453L006PASSBOB 0504 0191 TCPIP 090C058F06

When a user reaches the maximum number of attempts with a wrong password when trying to link a protected minidisk, this user cannot use the **LINK** command during the current session.

When the **Set_and_Query** parameter is set to on in SYSTEM CONFIG file, it is possible to control the Journaling by using the **set** CP command. Example 5-28 shows some examples of **query** and **set journal**.

Note: The SET JOURNAL command can be used only if the FACILITY ON operand and SET_AND_QUERY ON operand are specified on the JOURNALING statement in the system configuration file.

Example 5-28 Query and set journal

```
q journal
Journal: LOGON- on , LINK- on
Ready;
set journal logon off
Ready;
q journal
Journal: LOGON- off, LINK- on
Ready;
set journal link off
Ready;
q journal
Journal: LOGON- off, LINK- off
Ready;
set journal logon on
Ready;
q journal
Journal: LOGON- on , LINK- off
Ready;
set journal link on
Ready;
q journal
Journal: LOGON- on , LINK- on
Ready;
```

For more information about CP Journaling, see *CP Planning and Administration*, SC24-6271.

5.4.2 Auditing with RACF

Certain user roles or tasks are common to all users. At any installation, different users have different levels of responsibility for security or different needs to access resources. Some people might have extensive responsibility for security, and others might have little or none. Some users might require almost unlimited access to resources, and others might need only limited access. Some might be barred from entering the system at all.

The primary means of defining a user's responsibility for security is the RACF user attribute. The separation of powers is necessary because it is the security administrator's job to establish RACF controls, and it is the auditor's job to test the adequacy and effectiveness of these controls.

The auditor's responsibility is to verify that RACF is meeting the installation's security goals, such as access control and accountability. The job of a RACF auditor is essentially the same, regardless of whether it is the AUDITOR attribute (with responsibility for checking RACF controls on a user or system-wide, level) or the group-AUDITOR attribute (with responsibility for checking RACF controls for a group and its subgroups).

An effective audit of security goals depends on how the events are logged. Logging all the necessary information and events improves the effectiveness of an audit.

Enabling auditing

You can enable (audit) or disable (noaudit) functions dynamically to meet the needs of your installation. When you enable collection of the audit records, SMF records are generated. This optional step is part of the configuration of your RACF environment, as described in "Customizing the processing of SMF records" on page 72. If you elected not to perform that step previously, you must implement it now before continuing.

RACF always logs information about certain events that are essential to an effective data-security mechanism. RACF always logs the following events:

- ▶ Every use of the **RVARY** or **SETROPTS** command.
- ▶ Whenever a **RACROUTE REQUEST=VERIFY** request fails.
- ▶ Whenever the console operator grants access to a resource as part of the failsoft processing that is performed when RACF is inactive.

RACF never logs some events because knowing about these events is not essential to effective data security. RACF never logs any use of the following RACF commands:

- ▶ LISTDSD
- ▶ LISTGRP
- ▶ LISTUSER
- ▶ RLIST
- ▶ LDIRECT
- ▶ LFILE
- ▶ SRFILE
- ▶ SRDIR
- ▶ SEARCH

In addition to the events that RACF always logs and never logs, RACF can optionally log other events. Optional logging is under the control of a resource-profile owner or the auditor.

The first step in establishing the auditing environment is to activate the RACF class for auditing with the **SETROPTS** command. With this command, you specify what functions within the AUDIT facility you want to monitor (above what RACF always monitors). The following functions are included:

- ▶ USERS
- ▶ VMMDISK
- ▶ VMLAN
- ▶ VMRDR
- ▶ VMCMD
- ▶ VMNODES
- ▶ SURROGAT

Use the **SETROPTS LIST** command as a user with the AUDITOR attribute to determine your current AUDIT environment, as shown in Example 5-29 on page 137.

Example 5-29 AUDIT CLASS functions

```
rac lu edi
USER=EDI NAME=UNKNOWN OWNER=IBMUSER CREATED=19.167
DEFAULT-GROUP=SYS1 PASSDATE=19.167 PASS-INTERVAL= 30 PHRASEDATE=N/A
ATTRIBUTES=AUDITOR
REVOKE DATE=NONE RESUME DATE=NONE
LAST-ACCESS=16.174/15:00:22
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED (DAYS) (TIME)
-----
ANYDAY ANYTIME
GROUP=SYS1 AUTH=USE CONNECT-OWNER=IBMUSER CONNECT-DATE=19.167
CONNECTS= 06 UACC=NONE LAST-CONNECT=19.174/15:00:22
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
Ready; T=0.01/0.01 15:14:44
```

```
rac setropts list
ATTRIBUTES = INITSTATS NOWHEN(PROGRAM) SAUDIT CMDVIOL NOOPERAUDIT
STATISTICS = NONE
AUDIT CLASSES = NONE
ACTIVE CLASSES = DATASET USER GROUP VMMDISK VMRDR VMBATCH VMLAN VMSEGMT
                  FACILITY SURROGAT VXMBR VMXEVENT XFACILIT GXFACILI
GENERIC PROFILE CLASSES = NONE
GENERIC COMMAND CLASSES = NONE
GENLIST CLASSES = NONE
GLOBAL CHECKING CLASSES = NONE
RACLIST CLASSES = NONE
```

As shown in Example 5-29, auditing for RACF classes is not enabled. Before enabling any of the functions, you must start the RACFSMF VM and update the **PROFILE EXEC** for the AUTOLOG2 VM to start RACFSMF when the system is IPLed.

The following main utilities are used to manage the RACF generated SMF records in the z/VM environment:

- ▶ RACF Report Writer
- ▶ RACF SMF Data Unload

The Report Writer utility supports audit records for RACF 1.9.2 and earlier. It does not support most of the audit records that were introduced in RACF 1.10 for z/VM or later releases. RACF Report Writer requires the use of *tdisk* space on your system. You must discuss with your z/VM system programmer whether *tdisk* space was defined on your system. If it was not defined, it must be added.

These utilities are on the RACFVM 305 disk, and the disk must be linked and accessed before execution.

You start by turning on a few other AUDIT features on the z/VM system before running these programs. Enable AUDIT on classes on which you intend to log security events. An example is shown in Example 5-30.

Example 5-30 Enabling AUDIT

rac setropts audit (user group vmmdisk vmrdr vmlan surrogate)

```
OUTPUT FROM RACFVM ON SYSTEM ITS0ZVM3
ICH14004I UNABLE TO OPEN RACF DATA SET RACF.DATASET.
END OF OUTPUT FROM RACFVM ON SYSTEM ITS0ZVM3
OUTPUT FROM RACFVM ON SYSTEM ITS0ZVM2
ICH14004I UNABLE TO OPEN RACF DATA SET RACF.DATASET.
END OF OUTPUT FROM RACFVM ON SYSTEM ITS0ZVM2
Ready; T=0.01/0.01 15:24:28
```

rac setropts list

```
ATTRIBUTES = INITSTATS NOWHEN(PROGRAM) SAUDIT CMDVIOL NOOPERAUDIT
STATISTICS = NONE
AUDIT CLASSES = USER GROUP VMMDISK VMRDR VMLAN SURROGAT
ACTIVE CLASSES = DATASET USER GROUP VMMDISK VMRDR VMBATCH VMLAN VMSEGMT
                  FACILITY SURROGAT VXMBR VMXEVENT XFACILIT GFXACILI
GENERIC PROFILE CLASSES = NONE
GENERIC COMMAND CLASSES = NONE
GENLIST CLASSES = NONE
GLOBAL CHECKING CLASSES = NONE
RACLIST CLASSES = NONE
```

RACF Data Security Monitor Utility

RACF Data Security Monitor Utility (RACDSMON) is a program that produces reports on the status of the security environment of your installation and the status of resources that RACF controls. You can use the reports to audit the status of your installation's system security environment by comparing the system characteristics and resource-protection levels with the intended characteristics and levels. You can also control the reporting that RACDSMON does by specifying control statements that request certain functions for user input.

Before running the **RACDSMON EXEC**, the following requirements must be met:

- ▶ Have READ access to the RACF service's 305 and 490 minidisks and the primary and backup RACF databases.
- ▶ Have the AUDITOR attribute.
- ▶ Have at least 20 MB of virtual storage available for your user ID.
- ▶ Perform an IPL of the 490 disk.
- ▶ Access the 305 disk.

Perform the steps that are shown in Example 5-31 when you perform an IPL of the 490 disk. Depending upon what CMS commands are run from the **PROFILE EXEC**, you might receive some errors. You can disregard those error messages.

Example 5-31 Preparing to run RACDSMON EXEC

link racfvm 490 490 rr

```
DASD 0490 LINKED R/O; R/W BY RACFVM at ITS0ZVM4
Ready; T=0.01/0.01 15:37:47
```

link racfvm 305 305 rr


```
DASD 0305 LINKED R/O; R/W BY RACFVM    at ITS0ZVM4
Ready; T=0.01/0.01 15:38:12
```

ipl 490

```
RACFVM CMS XA Re1. 27 2011-10-18
Ready; T=0.01/0.01 15:38:50
```

acc 305 1

```
DMSACP723I L (305) R/O
Ready; T=0.01/0.01 15:39:28
```

acc 190 t

```
DMSACP723I T (190) R/O
Ready; T=0.01/0.01 11:42:47
```

During the example tests, the exec ran with some problems:

- In the example environment, no temp disk was available that was large enough to hold the same disk size of the RACF database. The environment was created with four SSI members, which means you must allocate a full pack DASD for the RACF database.

To overcome this situation, and knowing that the size that the RACF database uses is less than the full pack disk, create a smaller copy of the RACF database on a temporary disk by using DDR. Example 5-32 shows the output of this process.

Example 5-32 Creating a smaller copy of the RACF database

def t3390 200 100

```
DASD 0200 DEFINED
Ready; T=0.01/0.01 10:27:41
```

def t3390 300 100

```
DASD 0300 DEFINED
Ready; T=0.01/0.01 10:27:45
```

link racfvm 200 f200 rr

```
DASD F200 LINKED R/O; R/W BY RACFVM    at ITS0ZVM4
Ready; T=0.01/0.01 10:29:06
```

link racfvm 300 f300 rr

```
DASD F300 LINKED R/O; R/W BY RACFVM    at ITS0ZVM4
Ready; T=0.01/0.01 10:29:13
```

ddr

```
z/VM DASD DUMP/RESTORE PROGRAM
```

```
ENTER:
```

in f200 dasd

```
ENTER:
```

out 200 dasd

```
ENTER:
```

sys cons

```
ENTER:
```

copy 0 99

```
HCPDDR711D VOLID READ IS RACF
```

```
DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:
```

yes

```
ENTER NEXT EXTENT OR NULL LINE
```

ENTER:

HCPDDR716D NO VOL1 LABEL FOUND

DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:

yes

COPYING RACF

COPYING DATA 06/23/16 AT 14.32.50 GMT FROM RACF

INPUT CYLINDER EXTENTS OUTPUT CYLINDER EXTENTS

START	STOP	START	STOP
0	99	0	99

END OF COPY

ENTER:

END OF JOB

Ready; T=0.01/0.02 10:33:26

ddr

z/VM DASD DUMP/RESTORE PROGRAM

ENTER:

in f300 dasd

ENTER:

out 300 dasd

ENTER:

sys cons

ENTER:

copy 0 99

HCPDDR711D VOLID READ IS RACFBK

DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:

yes

ENTER NEXT EXTENT OR NULL LINE

ENTER:

HCPDDR716D NO VOL1 LABEL FOUND

DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:

yes

COPYING RACFBK

COPYING DATA 06/23/16 AT 14.34.08 GMT FROM RACFBK

INPUT CYLINDER EXTENTS OUTPUT CYLINDER EXTENTS

START	STOP	START	STOP
0	99	0	99

END OF COPY

ENTER:

END OF JOB

Ready; T=0.01/0.02 10:34:25

det f200

DASD F200 DETACHED

Ready; T=0.01/0.01 10:34:44

det f300

DASD F300 DETACHED

Ready; T=0.01/0.01 10:34:48

- The **RACDSMON EXEC** must be run while RACFVM 490 disk is running, but this program uses some utilities on CMS 190 disk. When 490 is running, 190 is not accessed. If the 190 minidisk is not accessed, you might receive the error messages that are shown in Example 5-33.

Example 5-33 Error messages when 190 disk is not accessed while running RACDSMON

Program ICHDSM00 is being executed - Please wait

```
223 +++ extents = C2D(Storage(load_address,1))      /* save RACF db xtnt
*/
DMSREX475E Error 40 running RACONFIG EXEC, line 223: Incorrect call to routine
```

```
An Error occurred during ICHDSM00 processing
Return code from ICHDSM00 = 20040
Ready(20040); T=0.01/0.01 10:38:17
```

To overcome this situation, after you perform an IPL of 490, access 190. The full output is shown in Example 5-31 on page 138.

- When **RACDSMON EXEC** runs, the generated output is placed in your virtual printer. You should run the **cp spool print *** command so that you can receive the files when this process completes. (You might want to add this command to your **PROFILE EXEC**.)

The RACFVM 200 and 300 disks in this case are the copy of the original locations of the RACF database. The RACDSMON generated reports are pulled from those disks. When the exec is run, it creates a tdisk of the same disk type as the 200 and 300 disks. You can issue the CP commands **query virtual 200** and **query virtual 300** to determine this information.

You can install the z/VM system on 3390 DASD or on a simulated 9330 Fixed Block SCSI disk. Therefore, one of these is the type of tdisk space that you must define. In this example, install the system on 3390 DASD.

To run the RACDSMON utility, enter the CP command **racdsmon**. It displays several panels. The first panels are only informational in nature. You can use one of them to go into a CMS SUBSET environment, where you can perform tasks such as linking to the disk that you should have linked before running the exec. The first panels where you must provide information is the panel that prompts you for the address of the INPUT RACF database device (see Example 5-34). When the exec runs, these prompts are displayed on three separate panels.

Example 5-34 RACF database input

Enter the INPUT RACF dataset device address one at a time.

Enter END when all input data sets are entered.
or

Enter QUIT to terminate processing.

200

Enter the NEXT INPUT RACF dataset device address .

Enter END when all input data sets are entered.
or

Enter QUIT to terminate processing.

300

Enter the NEXT INPUT RACF dataset device address .

Enter END when all input data sets are entered.

or

Enter QUIT to terminate processing.

end

-
- The next prompt from the exec asks whether you want to use a tdisk or a minidisk. If your system does not have tdisk space that is defined, then you can use minidisks. These disks must be defined in the system directory and must be the same size and geometry as the 200 and 300 disk that is owned by RACFVM. In our example, we use tdisk, as shown in Example 5-35. Then, you see panels that display messages about the copy of the 200 and 300 disks to the 5FD and 5FE disks.

Example 5-35 Use of temporary disk

DMSACC723I H (0200) R/W - OS

Would you like to use TDISK or existing disks for the file to scan?

Enter "T" to TDISK or "E" for existing disk(s)

T

The input RACF data set - 200 is being copied over to 5FE

...Please wait...

DMSACC724I 300 replaces H (200) - OS

DMSACC723I H (0300) R/W - OS

The input RACF data set - 300 is being copied over to 5FD

...Please wait...

-
- You should receive a message about the ICHDSM00 SYSIN file and have an opportunity to edit the file (see Example 5-36). If it is not the first time you are running RACDSMON on this machine, you receive a message saying that the file exists and if you want to overlay it. Answering YES deletes the file on the disk and creates a new file. Accept the default on this panel, and edit the file.

Example 5-36 ICHDSM00 SYSIN file message

The ICHDSM00 SYSIN file will initially contain all DSMON FUNCTION control statements that are applicable to VM .

XEDIT will be invoked in order to tailor the ICHDSM00 SYSIN file.

Please be sure to issue the FILE command when edits are completed.

Press Enter to go into XEDIT

You must modify the ICHDSM00 SYSIN file. It features one option that is not supported on RACF for z/VM. Example 5-37 on page 143 shows the correct modifications for this file. Save the file after you modify it. When running RACDSMON in the future, you can respond to the question about editing this file with NO (which then uses the file on your A disk).

Example 5-37 Updating ICHDSM00 SYSIN

```
===== * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== FUNCTION SYSTEM
===== FUNCTION RACGRP
===== FUNCTION RACCDT
==d== FUNCTION RACEXT
===== FUNCTION RACGAC
===== FUNCTION RACUSR
===== FUNCTION RACDST
===== * * * End of File * * *
```

After a few minutes, you receive the messages that are shown in Example 5-38.

Example 5-38 RACDSMON completion

Program ICHDSM00 is being executed - Please wait

```
DMSACC723I R (0200) R/W - OS
DMSACC723I Q (0300) R/W - OS
CSTSET001I CMS SUB-TASKING SUPERVISOR INITIALIZED.
CSTINT003I INITIATOR ACTIVATED.
PRT FILE 0012 SENT FROM EDI PRT WAS 0012 RECS 0352 CPY 001 A NOHOLD NOKEEP
CSTINT004I PROGRAM 'RACFIPLU' ENDED. COMPLETION CODE = 000000.
CSTINT006I NO MORE SUB-TASKS.
CSTTER001I CST TERMINATED.
```

```
Return code from ICHDSM00 = 0
PRT FILE 0016 SENT FROM EDI PRT WAS 0016 RECS 0026 CPY 001 A NOHOLD NOKEEP
Ready; T=0.04/0.05 11:15:12
```

When the **RACDSMON EXEC** completes, you must perform an IPL of the CMS or 190 to have full CMS function. The 490 disk that is owned by RACFVM does not include all of the CMS executable files that you have with a normal CMS. After performing the IPL of the 490 disk, you do not have access to FIELIST, RDRLIST, FULLIST, XEDIT, and so on.

The **RACDSMON EXEC** creates two files in your VMRDR. The file that is named (none) is the report that is generated. You should receive a file that provides a file name and file type that is used by the security audit team. You can discard the file ICHDSM00 \$\$\$\$\$\$. If you forgot to spool the VMPRT, transfer the files from the VMPRT to the VMRDR.

The audit report includes the following information:

- RACF System Report (see Example 5-39).

Example 5-39 RACF System Report

```
RACF DATA SECURITY MONITOR
```

	S Y S T E M	R E

CPU-ID	ODDA87	
CPU MODEL	2964	
OPERATING SYSTEM/LEVEL	z/VM Version 7 Release 1.0, service 1	
LAST SYSTEM GENERATION	Generated at 06/16/19 12:07:28 EDT	
LAST SYSTEM IPL	IPL at 06/22/19 16:57:27 EDT	
RACF VERSION 7 RELEASE 1 IS ACTIVE		
RACF DATA SECURITY MONITOR		

- RACF Exits Report (see Example 5-40).

Example 5-40 RACF Exits Report

```

DSMON    RPT0723  A1  F 132  Trunc=132 Size=356 Line=1
====>
  11      R A C F      E X I T S      R E P O R T
  12 EXIT MODULE          MODULE
  13 NAME                LENGTH
  14 -----
  15 ICHPWX11            1,520
  16 RACF DATA SECURITY MONITOR

```

- Selected User Attribute Report (see Example 5-41).

Example 5-41 Selected User Attribute Report

S E L E C T E D	U S E R	A T T R		
USERID	ATTRIBUTE TYPE			
	SPECIAL	OPERATIONS	AUDITOR	REVOKE

\$ALLOC\$				SYSTEM
\$DIRECT\$				SYSTEM
\$PAGE\$				SYSTEM
\$SPOOL\$				SYSTEM
\$SYSCKP\$				SYSTEM
\$SYSWRM\$				SYSTEM
\$TDISK\$				SYSTEM
BLDCMS		SYSTEM		
BLDNUC		SYSTEM		
BLDRACF		SYSTEM		
BLDSEG		SYSTEM		
IBMUSER	SYSTEM	SYSTEM		
KLAUSM	SYSTEM			
MAINT	SYSTEM			
MAINT630		SYSTEM		
MIGMAINT		SYSTEM		
RAMPAZZO		SYSTEM	SYSTEM	
VIC	SYSTEM			
WILLIANR	SYSTEM	SYSTEM		
6VMLEN20				SYSTEM
RACF DATA SECURITY MONITOR				
S E L E C T E D U S E R A T T R I B U T				

TOTAL DEFINED USERS:		156		
TOTAL SELECTED ATTRIBUTE USERS:				
ATTRIBUTE BASIS	SPECIAL	OPERATIONS	AUDITOR	

SYSTEM	5	9		
GROUP	0	0		

- RACF Class Descriptor Table Report (see Example 5-42).

Example 5-42 RACF Class Descriptor Table

R A C F CLASS NAME	C L A S S STATUS	D E S C R I P T O AUDITING	STATISTICS	DEFAULT UACC
RVARSMBR	INACTIVE	NO	NO	NONE
RACFVARS	INACTIVE	NO	NO	NONE
SECLABEL	INACTIVE	NO	NO	NONE
VMMDISK	ACTIVE	YES	NO	NONE
VMRDR	ACTIVE	YES	NO	NONE
VMCMD	INACTIVE	NO	NO	NONE
VMNODE	INACTIVE	NO	NO	NONE
VMBATCH	ACTIVE	NO	NO	NONE
VMDEV	INACTIVE	NO	NO	NONE
FILE	INACTIVE	NO	NO	NONE
DIRECTRY	INACTIVE	NO	NO	NONE
SFSCMD	INACTIVE	NO	NO	NONE
VMPOSIX	INACTIVE	NO	NO	NONE
VMLAN	ACTIVE	YES	NO	NONE
VMMAC	INACTIVE	NO	NO	NONE
VMSEGMT	ACTIVE	NO	NO	NONE

- RACF Global Resource Table Report (see Example 5-43).

Example 5-43 RACF Global Resource Table

R A C F CLASS NAME	G L O B A L ACCESS LEVEL	A C C E S S ENTRY NAME
DATASET		-- GLOBAL INACTIVE --
RVARSMBR		-- GLOBAL INACTIVE --
SECLABEL		-- GLOBAL INACTIVE --
VMMDISK		-- GLOBAL INACTIVE --
VMRDR		-- GLOBAL INACTIVE --
VMCMD		-- GLOBAL INACTIVE --
VMNODE		-- GLOBAL INACTIVE --
VMBATCH		-- GLOBAL INACTIVE --
VMDEV		-- GLOBAL INACTIVE --
FILE		-- GLOBAL INACTIVE --
DIRECTRY		-- GLOBAL INACTIVE --
SFSCMD		-- GLOBAL INACTIVE --
VMPOSIX		-- GLOBAL INACTIVE --
VMLAN		-- GLOBAL INACTIVE --
VMMAC		-- GLOBAL INACTIVE --
VMSEGMT		-- GLOBAL INACTIVE --

- RACF Group Tree Report (see Example 5-44).

Example 5-44 RACF Group Tree Report

R A C F	G R O U P	T R E E
LEVEL	GROUP	(OWNER)

1	SYS1	(IBMUSER)
2	DIRMADMN	
2	DIRMSRV	
2	GADM	(IBMUSER)
2	GBIN	(IBMUSER)
2	GNOBODY	(IBMUSER)
2	GSYS	(IBMUSER)
2	MAIL	(IBMUSER)

RACF SMF Data Unload Utility

Available since RACF/VM 1.10 and RACF FL 530, the RACF SMF Data Unload Utility (RACFADU) is the IBM preferred utility for processing RACF audit records. With it, you can create a sequential file from the security relevant SMF data. You can use the sequential file in the following ways:

- View the file directly.
- Use the file as input for installation-written programs.
- Manipulate the file with sort and merge utilities.
- Output to an XML-formatted file for viewing with a web browser.

If the output is loaded into a database management system (for example, IBM DB2® or SQL/DS), you can issue your own queries. RACF includes the sample statements that are required to define and load the IBM Db2® tables.

Before you can run the **RACFADU EXEC**, the following requirements must be met:

- Link and access the RACFVM 305 disk.
- Link the RACFVM 301 and 302 disks.
- Have adequate free space on your A disk for the output file (30 cylinders is acceptable).

You can run **RACFADU EXEC** with or without any parameters. Without any parameters, it opens the RACFADU panel (see Example 5-45 on page 147).

Example 5-45 RACF SMF Unload Utility

RACF SMF Unload Utility - Input Panel

. Virtual address of input SMF data minidisk _____

. Virtual address of output minidisk _____

. Filename and filetype of sequential output file RACFADU OUTPUT

. Filename and filetype of XML easily readable output file _____

. Filename and filetype of XML compressed output file _____

PF1 = Help PF2 = Execute PF3 = Quit
ENTER = Verify input fields

Enter CP/CMS Commands below:
====>

Example 5-46 shows the command that is run with all the required options for the command, which bypasses the input panel.

Example 5-46 RACFADU without an input panel

```
racfadu 301 191
RACFADU OUTPUT
RPIADU033I SMF unload completed successfully.
View the RACFADU MESSAGES file for additional details.
Ready; T=0.01/0.01 13:32:28
```

When you run the exec in either mode, the following files are created on your A disk by default:

- ▶ RACFADU MESSAGES A1
- ▶ RACFADU OUTPUT A1

The RACFADU MESSAGES file describes how many of each type of SMF records were processed, as shown in Example 5-47.

Example 5-47 RACFADU MESSAGES file

```
* * * Top of File * * *
IRR67652I The utility processed 0 SMF type 30 records.
IRR67652I The utility processed 223 SMF type 80 records.
IRR67652I The utility processed 5 SMF type 81 records.
IRR67655I The utility processed 0 SMF type 83 subtype 1 records.
IRR67655I The utility processed 0 SMF type 83 subtype 2 records.
IRR67655I The utility processed 0 SMF type 83 subtype 3 records.
IRR67655I The utility processed 0 SMF type 83 subtype 4 records.
IRR67655I The utility processed 0 SMF type 83 subtype 5 records.
IRR67655I The utility processed 0 SMF type 83 subtype 6 records.
IRR67653I The utility bypassed 0 SMF records not related to IRRADU00.
```

IRR67650I SMF data unload utility has successfully completed.
* * * End of File * * *

The RACFADU OUTPUT file is the readable output of all the SMF data records (see Example 5-48). If this file exists on the output disk, you are prompted to rename or replace the old file before continuing. These files can be used by Db2, SQL/DS, and DFSORT/CMS.

Example 5-48 RACFADU OUTPUT file

```
RACFADU  OUTPUT  A1  V 5331  Trunc=5331 Size=5043 Line=3972 Col=1 Alt=0
====>
3972 DEFINE      SUCCESS  11:00:34 2007-07-18 VMSP NO    NO    NO    DETRO  SYS1
3973 RDEFINE     SUCCESS  11:00:34 2007-07-18 VMSP NO    NO    NO    DETRO  SYS1
3974 PERMIT      SUCCESS  11:00:34 2007-07-18 VMSP NO    NO    NO    DETRO  SYS1
3975 PERMIT      SUCCESS  11:00:34 2007-07-18 VMSP NO    NO    NO    DETRO  SYS1
3976 PERMIT      SUCCESS  11:00:34 2007-07-18 VMSP NO    NO    NO    DETRO  SYS1
3977 PERMIT      SUCCESS  11:00:34 2007-07-18 VMSP NO    NO    NO    DETRO  SYS1
3978 PERMIT      SUCCESS  11:00:34 2007-07-18 VMSP NO    NO    NO    DETRO  SYS1
3979 PERMIT      SUCCESS  11:00:34 2007-07-18 VMSP NO    NO    NO    DETRO  SYS1
3980 ACCESS      SUCCESS  11:11:44 2007-07-18 VMSP NO    NO    NO    DETRO  SYS1
4598 ACCESS      SUCCESS  17:18:19 2007-07-18 VMSP NO    NO    NO    DETRO  SYS1
4751 ACCESS      SUCCESS  17:24:04 2007-07-18 VMSP NO    NO    NO    DETRO  SYS1
```

The utility can be used to generate an XML file that is readable with a browser. To create the XML file, you must pass the file name and file type for the XML file. The following parameters are included:

- ▶ OUTXRN filename File name of output XML easily readable file.
- ▶ OUTXRT filename File type of output XML easily readable file.
- ▶ OUTXCN filename File name of output XML compressed.
- ▶ OUTXCT filename File type of output XML compressed.

It is much easier to use the panel when the XML files are generated from your SMF data (see Example 5-49 on page 149). Also, if you are using the XML function, it works better with the compressed version of this process.

Example 5-49 Using the RACF SMF Unload Utility Input panel to generate XML files from SMF data

RACF SMF Unload Utility - Input Panel

```
. Virtual address of input SMF data minidisk      0301

. Virtual address of output minidisk              0191

. Filename and filetype of sequential
  output file

. Filename and filetype of XML easily readable   _____
  output file

. Filename and filetype of XML compressed        RACFADU1 XML_____
  output file
```

PF1 = Help PF2 = Execute PF3 = Quit
ENTER = Verify input fields

Enter CP/CMS Commands below:

====>

After the file is created, use the IBM Personal Communications program or any other method available on your installations to download the file from the A disk to the desktop in *text* mode. After you download the file, open it with an editor and change the encoding value and XML syntax error on the first line, as shown in Example 5-50 (and as documented in *RACF Security Server Auditor's Guide*, SC24-6311). This step is required because of the mismatch between the z/VM use of EBCDIC versus the PC that uses ASCII.

Example 5-50 Changing the XML header

```
<xml version='1.0' encoding='ISO8859-1'>
<securityEventLog xmlns='http://www.ibm.com/xmlns/zOS/IRRSchema'>

  <rdf:Description rdf:about=''
                    xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
                    xmlns:dc='http://purl.org/dc/elements/1.1/'>
    <dc:creator>      RACF for z/VM      SMF Unload (HRF6030)</dc:creator>
    <dc:subject>RACF Security Event Log 2016-06-23 13:37:39</dc:subject>
    <dc:language>en</dc:language>
  </rdf:Description>
```

At the bottom of the file, it is missing the close tag for the XML, and some browsers might encounter problems opening the file. In this case, add the missing tag, as shown in Example 5-51.

Example 5-51 Correcting the XML close tag

```
</securityEventLog>
</xml>
```

You can view the audit report on personal computers and workstations by using an XML-capable web browser. Many browsers that are available today can correctly parse and render XML documents. Therefore, when the audit report is on that system, you can read it as easily as any other web document. Open a listing of the files and single- or double-click the file to open it in the browser window.

In this example, when this file is opened by using Firefox, you receive a message in reference to a missing style file (see Figure 5-2). In this case, you must combine this file with a customized stylesheet to get the browser to filter and display the windows in a more acceptable format.

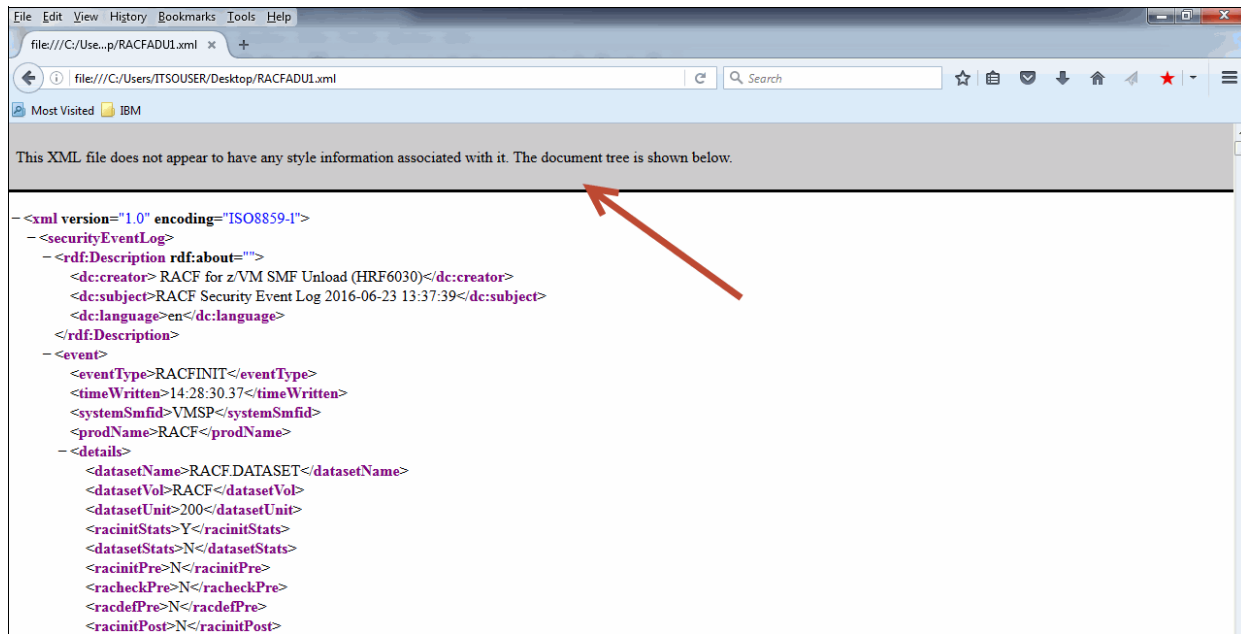


Figure 5-2 Opening an XML file with Firefox

Internet Explorer opens the file without any message, as shown in Figure 5-3 on page 151.

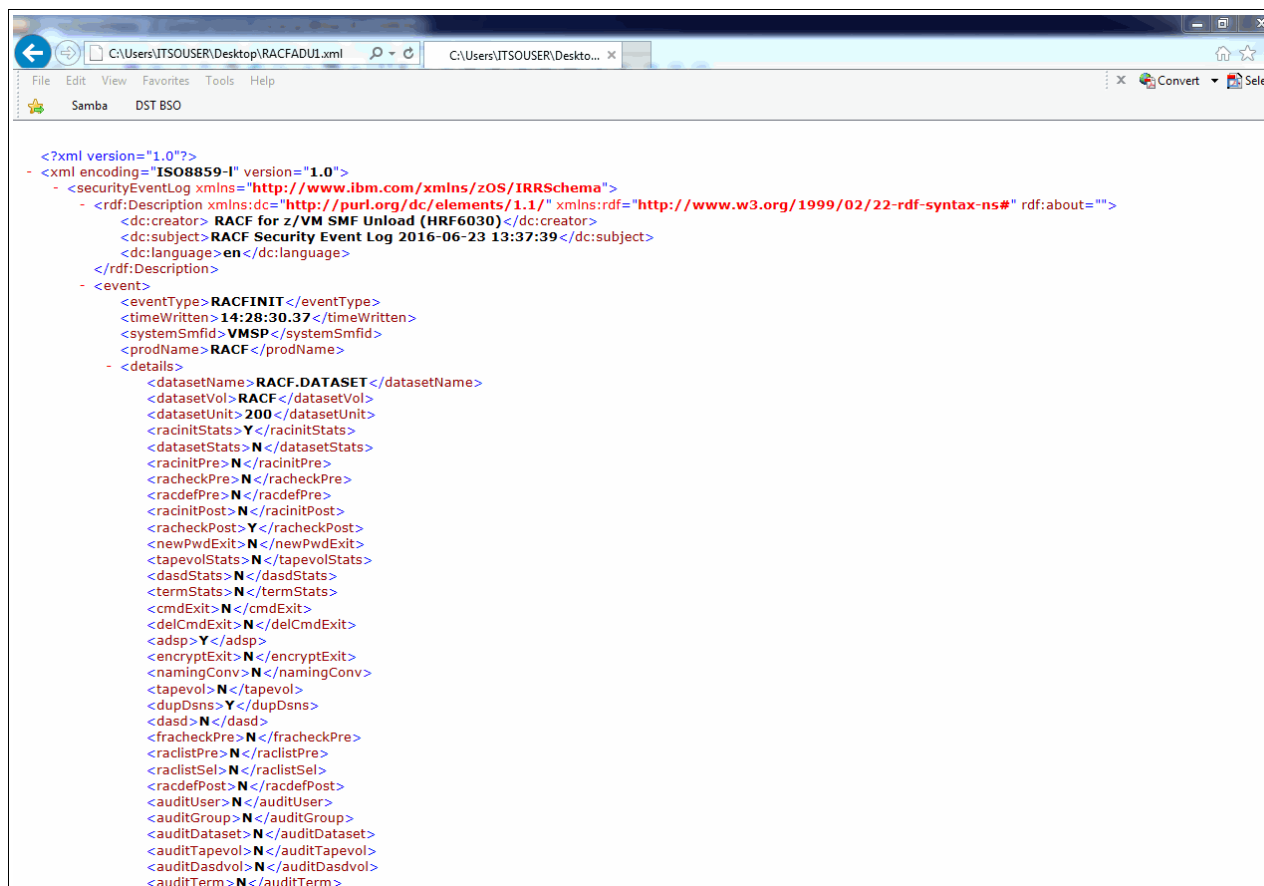


Figure 5-3 Opening an XML file with Internet Explorer

RACF Report Writer Utility

Note: The RACF Report Writer Utility (RACFRW) is no longer the IBM preferred utility for processing RACF audit records. The RACF SMF data unload utility is the preferred reporting utility. The RACFRW does not support many of the audit records that are introduced after RACF 1.9.2.

The RACFRW lists the contents of System Management Facilities (SMF) records in a format that is easy to read. SMF records are in the SMF data file. You can also tailor the reports to select specific SMF records that contain certain kinds of RACF information. With the RACFRW, you can obtain the following information:

- ▶ Reports that describe attempts to access a particular RACF-protected resource in terms of user identity, number and type of successful accesses, and number and type of attempted security violations.
- ▶ Reports that describe user and group activity.
- ▶ Reports that summarize system use and resource use.

The RACF report writer lists the contents of SMF records in a format that is easy to read. It provides a wide range of reports so that you can monitor and verify the use of the system and resources.

The RACF report writer consists of the following phases:

1. Command and subcommand processing.
2. Record selection.
3. Report generation.

The steps to perform when running the **RACRPORT EXEC** are similar to the steps in running other RACF utilities. You must link and access several disks that are owned by RACFVM and then run the exec (see Example 5-52). Unlike the RACDSMON where you must link to the RACFVM 200 and 300 disks (the location of the RACF database), this time you need access to the SMF records that are created on the RACFVM 301 and 302 disks.

Example 5-52 Linking the necessary disks for RACRPORT

```
link racfvm 191 291 rr  
DASD 0291 LINKED R/O; R/W BY RACFVM    at ITS0ZVM4  
Ready; T=0.01/0.01 14:22:22
```

```
link racfvm 301 301 rr  
DASD 0301 LINKED R/O; R/W BY RACFVM    at ITS0ZVM4  
Ready; T=0.01/0.01 14:22:29
```

```
link racfvm 302 302 rr  
Ready; T=0.01/0.01 14:22:36
```

```
link racfvm 305 305 rr  
DASD 0305 LINKED R/O; R/W BY RACFVM    at ITS0ZVM4  
Ready; T=0.01/0.01 14:22:42
```

```
acc 305 1  
DMSACP723I L (305) R/O  
Ready; T=0.01/0.01 14:41:36
```

While working with RACRPORT, we observed that the tdisk that was used by this utility was hardcoded in the program with virtual address 5FF. If you use that virtual address for something else, you must detach it or redefine the disk to another address.

The **RACRPORT EXEC** generates the reports in your virtual printer. Therefore, spool your printer to your reader to make things easier.

The **SET SECUSER** command is used to change the secondary user ID that is associated with your VM or another user's VM. To run the command for the OPERATOR VM, you must be authorized to run class A privileged commands (see Example 5-53 on page 153). This command is the command that additionally authorizes the use of the CP **SEND** command.

Example 5-53 CP query privclass

```
id
EDI AT ITS0ZVM4 VIA *          06/23/16 14:26:59 EDT    THURSDAY
Ready;
  q priv
Privilege classes for user EDI
    Currently: ABCDEFG
    Directory: ABCDEFG
The privilege classes are not locked against changes.
```

Before you can run this utility, RACFVM must switch from the primary SMF disk to the alternative SMF disk. This task is accomplished by using the RACF **SMF SWITCH** command.

During the RACF implementation, an optional step was available to change the Message Routing Table, which allowed you to define more VMs that are allowed to request the SMF SWITCH of RACFVM. At that time, you did not need to add VMs to this list. Now, you implement this process by having the OPERATOR VM issue the **SMF SWITCH**, as shown in Example 5-54.

Example 5-54 Switching the SMF disk

```
q secuser racfvm
      Secondary
Userid  Userid  Status
RACFVM  OPERATOR logged on
Ready;
  set secuser racfvm *
HPCPCFX6768I SECUSER of RACFVM initiated.
Ready;
  msg racfvm smf switch
You are not authorized to issue MSG to a RACF server
Ready;
  set secuser racfvm reset
RACFVM : HPCPCFX6768I Your SECUSER set to OPERATOR by EDI.
HPCPCFX6769I SECUSER of RACFVM terminated.
Ready;
q secuser operator
      Secondary
Userid  Userid  Status
OPERATOR          not defined
Ready;
  set secuser operator *
HPCPCFX6768I SECUSER of OPERATOR initiated.
Ready;
  send cp operator msg racfvm smf switch
Ready;
  OPERATOR: RPISMF066I Switched to secondary disk
set secuser operator reset
OPERATOR: HPCPCFX6769I Your SECUSER terminated by EDI.
HPCPCFX6769I SECUSER of OPERATOR terminated.
Ready;
```

Note: As you can see in Example 5-54, OPERATOR and RACFSMF are the only VMs that are authorized to send messages to RACFVM.

If RACFSMF was not granted the authority to link to the RACFVM 301 and 302 disks, **SMF SWITCH** fails. The solution is to issue RACF **PERMITS** commands for those disks:

```
rac permit racfvm.301 class(vmmdisk) id(racfsmf) ac(control)
rac permit racfvm.302 class(vmmdisk) id(racfsmf) ac(control)
rac permit racfvm.191 class(vmmdisk) id(racfsmf) ac(read)
```

Now, when you run the **SMF SWITCH** command through the CP **SEND** command, it is successful. With this step completed, you can run the **RACRPORT** command after you have accessed the 305 disk (see Example 5-55).

Example 5-55 Running RACRPORT

The RACFRW CONTROL file cannot be located and is required for execution to continue.

Do you wish to create a RACFRW CONTROL file?
Please enter YES or NO

yes

XEDIT will be invoked in order to tailor the RACFRW CONTROL file.

Please be sure to issue the FILE command when edits are completed.
Please press Enter to continue

The RACFRW CONTROL file must contain the input that is required by RACFRW, including the **RACFRW** command and subcommands. This file is on your A disk and is created with XEDIT when **RACFRW** is run. The statements that are included in Example 5-56 generate a detailed report.

Example 5-56 RACFRW CONTROL file

```
RACFRW  CONTROL  A1  F 80  Trunc=80 Size=7 Line=7 Col=1 Alt=7
DMSXMD587I XEDIT:
===== * * * Top of File * * *
===== RACFRW TITLE ('PLACE YOUR RACF REPORT TITLE HERE')
===== SELECT VIOLATIONS
===== SELECT SUCCESSES
===== EVENT LOGON
===== EVENT SETROPTS
===== LIST
===== END
      |...+...1...+...2...+...3...+...4...+...5...+...
===== * * * End of File * * *
```

After modifying the RACFRW CONTROL file and saving it to your A disk, you are prompted to define where the work disk is stored (a tdisk or your A disk). Because the A disk is usually a small disk, use the tdisk. Respond to the prompt as shown in Example 5-57.

Example 5-57 Use of tdisk when running RACRPORT

The RACF Report Writer requires Disk Space for a Sort work file.
You may wish to use Tdisk for this function.
Note: If Tdisk is not used, the Sort work file will be written on the A disk.
Do you wish to use Tdisk for the Sort work file?
Please enter YES or NO
YES

The **RACRPORT EXEC** does not contain the logic that is in the **RACDSMON** exec, where **RACDSMON** might determine what type of tdisk space was defined on your system. With the **RACRPORT EXEC**, you must specify the tdisk disk type.

The **query tdisk** command (see Example 5-58) gives you information about system-defined tdisk space. However, it does not provide the characteristics of the disk device. You must query the devices to obtain the disk type.

Example 5-58 The query tdisk command

q tdisk

```
DASD 3D07 ATTACHED CPVOL 0000 VM3D07
DASD MDISKS NOT FOUND
```

```
DASD 3E07 ATTACHED CPVOL 0001 VM3E07
RAMPAZZO 05FF 00000007
Ready;
```

q 3d07 id

```
DASD 3D07 3390-0A CU: 2107-E8
      3D07 EQID: 002107900IBM750000000000DVV610D07000000000000D0A
Ready;
```

With this information, you can select the correct type of tdisk to create as the sort disk. The EXEC then provides you with the information about the SMF disk that is being used for the generation of this report (see Example 5-59). The requirement is that the sort disk must be the same size or larger than the source disk. As a preferred practice, make the sort disk the same size as the source disk.

Example 5-59 Information about rgw disk that contains the SMF DATA file

You will now be prompted for Tdisk space

Since the number of cylinders or blocks required depends on the size of the SMF DATA file, it is recommended that you allocate a temporary disk that is at least as large as the SMF DATA file.

The disk containing the SMF DATA file will be displayed below

LABEL	VDEV	M	STAT	CYL	TYPE	BLKSZ	FILES	BLKS USED-(%)	BLKS LEFT	BLK TOTAL
RCF301	301	C/A	R/O	7	3390	4096	1	33-03	1227	1260

Please enter the number of cylinders or blocks for Tdisk allocation.

7

After you define the size of the source disk, the tdisk is created as address 5FF (this address must be available). It then uses the definitions that you created in the **RACFRW CONTROL** file and generates a console file in your **VMRDR**. You can use the **peek** command to view this file or receive it to disk. If you print it, it should be printed on a printer that supports logical record lengths of 132 characters.

Because of variations from one installation to another, and all different types of policies that are used by the companies, it is not possible to identify all of the ways an auditor might use the **RACFRW**.

However, the following monitoring possibilities are available, which are described in *RACF Security Server Auditor's Guide*, SC24-6305:

- ▶ Password violation levels
- ▶ Access attempts in WARNING mode
- ▶ Access violations
- ▶ Use of RACF commands
- ▶ Specific users
- ▶ SPECIAL users
- ▶ OPERATIONS users
- ▶ Failed accesses to resources protected by a security level
- ▶ Accesses to resources protected by a security label



Securing a cloud in an IBM z/VM environment

Today's security requires consistent protection against threats and malware. Enterprises must be flexible while having a secure infrastructure to protect effectively the most valued asset of a company (the data), and their access through the cloud.

Running many distributed servers involves much effort to install, manage, maintain, and provide security for them. To contain this effort, many enterprises are consolidating these servers on IBM LinuxONE by using z/VM as the hypervisor, which takes advantage of the virtualization technologies to use the hardware effectively and to simplify administration tasks.

Implementing the enterprise security policy and following the least privilege principle increases the strength of security in your enterprise cloud.

This chapter describes the security of a Cloud on z/VM environment with its building blocks: z/VM Directory Manager (DirMaint), SMAPI, Extreme Cloud Administration Toolkit (xCAT), and Cloud Manager Appliance.

This chapter includes the following topics:

- ▶ 6.1, "Cloud on z/VM components" on page 158
- ▶ 6.2, "DirMaint" on page 159
- ▶ 6.3, "Systems Management API" on page 167
- ▶ 6.4, "z/VM Cloud Manager Appliance" on page 174
- ▶ 6.5, "CMA Controller node" on page 178
- ▶ 6.6, "CMA compute node" on page 182
- ▶ 6.7, "CMA installation" on page 184
- ▶ 6.8, "Securing your cloud components" on page 190

6.1 Cloud on z/VM components

An enterprise cloud might be composed of various components, depending on its main purpose. Implementing an Infrastructure as a Service (IaaS) cloud in z/VM demands the integration of some important components. The following components play a role in the integration of a cloud in z/VM:

- ▶ The z/VM Directory Manager (DirMaint) or a supported equivalent software provides a command-driven interface to manage z/VM directory entries.
- ▶ The z/VM Systems Management Application Programming Interface (SMAPI) provides programmatic access for managing many virtual images that are running on a single z/VM image by using a standard, platform-independent client interface, which reduces the number of z/VM-specific programming skills that are required.
- ▶ An External Security Manager (ESM) (such as IBM Resource Access Control Facility [RACF]) provides more resource protection beyond DIRMAINT and SMAPI authorizations. An ESM is optional, but implementing it might ensure that the company security policy is met. For more information about how to implement RACF, see Chapter 4, “IBM Resource Access Control Facility Security Server for IBM z/VM” on page 55.
- ▶ A Virtual Switch (VSWITCH) provides network connectivity between the management components to allow command-driven requests to come from the z/VM platform or other network -connected locations. They also provide the networks on to which newly provisioned instances are connected.
- ▶ The z/VM Cloud Manager Appliance (CMA) that is available is version 6.4 and is based on Newton. It provides an easy method to deploy z/VM OpenStack enablement. OpenStack products and solutions can be constructed to use as many or as few of the services as needed, whether that means that the CMA runs cloud controller services, compute node services, or only services that are needed by OpenStack z/VM drivers running in other virtual machines (VMs) or on other platforms.

Note: The CMA description in this publication that refers to CMA 6.4 (Newton) and the support for the Cloud Manager Appliance (CMA) on z/VM v7.1 is a transitional offer until a strategic long-term solution to replace the CMA is available. For more information, see IBM Cloud Private documentation at [this web page](#).

The Cloud Manager Appliance 6.4 is available for z/VM v7.1 for the specific use case with IBM Cloud Private only. For more information, contact your IBM representative.

The xCAT MN and the ZHCP server now both run within the OPNCLOUD virtual machine.

z/VM supports the following system roles, which control the set of services that are running inside the OPNCLOUD virtual machine:

- ▶ CMA controller
Runs cloud controller services (such as glance image services) and all services that are listed under the compute_mn role. z/VM also runs the xCAT MN and ZHCP services to allow the controller to manage OpenStack z/VM hosts.
- ▶ CMA compute
Runs compute services (nova-compute service), networking services (neutron-zvm-agent service), and telemetry services (ceilometer-polling) for the z/VM hypervisor. z/VM also runs the ZHCP service to allow a remote xCAT MN service to manage hosts.

- `Compute_MN`

Runs compute, networking, and telemetry services (listed under the CMA compute role) for the z/VM hypervisor. It also runs xCAT and ZHCP services.

This type of node is used in an environment where OpenStack controller services are run on a non-CMA node. The xCAT MN and ZHCP services allow a controller to manage the z/VM host without requiring cloud controller services to be running on the host.

- `mn`

Runs the xCAT and ZHCP services. This role is useful when all OpenStack services are running in non-CMA nodes or when you want to use xCAT and not OpenStack.

- `zhcp`

Runs only the ZHCP service. This role is useful when all OpenStack services are running in non-CMA nodes or when you want xCAT and not OpenStack. In this case, another z/VM host must run xCAT MN service to manage the host through the ZHCP service.

Note: Running the xCAT MN and the ZHCP server in separate virtual machines is not supported in z/VM 6.4 and later releases.

6.2 DirMaint

DirMaint provides well-organized and secure interactive facilities for maintaining the z/VM system directory. Directory management is simplified by the DirMaint command interface and automated facilities. DirMaint supports all the z/VM directory statements. Most DirMaint directory commands include the same names and format as the z/VM directory.

6.2.1 DirMaint controls

Important control files of DIRMAINT are available that can be tailored to work correctly according to your environment settings. These DIRMAINT files are described next.

When performing a new implementation of the product, you must modify or create the following files:

- `CONFIG DATADVH` (included with the product)
- `CONFIGnn DATADVH` (must be created)
- `AUTHFOR CONTROL` (must be updated)
- `RPWLST DATA` (copied from MAINT's 2CC disk)
- `EXTENT CONTROL` (must be updated)

CONFIG DATADVH

The `CONFIG DATADVH` file is the most important file for the DirMaint configuration. It contains all of the tailorable parameters for the product. You should *never* modify this IBM supplied file because it might be updated through the service process and overwrite your changes.

To change parameters on this file, create an override file instead where your installation-specific parameters are defined. The override file is named `CONFIGnn DATADVH`, where the *nn* can be any two digits you want to assign and is loaded in alphabetical order or numerical order.

For example, you can create CONFIGAA as your specifics settings and then DIRMAINT loads it after load CONFIG DATADVH. The CONFIG DATADVH file is self-documented. For more information, see *Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283. Example 6-5 on page 166

CONFIGSM DATADVH

Note: The CONFIGSM DATADVH file must be on the 11F disk when it is created. It is updated by using the **dirm send** and **dirm file** commands to get and replace this file. After changes are made, the DirMaint administrator must run the **dirm rldata** command. This command is used to instruct the DIRMAINT VM to reload all DATADVH files into memory.

The statements that are shown in Example 6-1 are considered as a good starting point for your CONFIGSM DATADVH file. DirMaint accepts *nn* as AA, BB, and so on. This example uses SM to refer to SMAPI configuration.

Example 6-1 CONFIGSM DATADVH

```
ALLOW_ASUSER_NOPASS_FROM= VSMGUARD *
ALLOW_ASUSER_NOPASS_FROM= VSMWORK1 *
ALLOW_ASUSER_NOPASS_FROM= VSMWORK2 *
ALLOW_ASUSER_NOPASS_FROM= VSMWORK3 *
ALLOW_ASUSER_NOPASS_FROM= WAVEWRKS *
ALLOW_ASUSER_NOPASS_FROM= WAVEWRKC *
ALLOW_ASUSER_NOPASS_FROM= WAVEWRKL *
ALLOW_ASUSER_NOPASS_FROM= EDI *
ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.TCP= DVHXNE EXEC
ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.UDP= DVHXNE EXEC
```

This file *must* be on the same disk as the CONFIG DATADVH file.

Note: Consider the following points:

- ▶ The ALLOW_ASUSER_NOPASS_FROM lines allow SMAPI users to issue commands to the Directory Manager by using the **ASUSER** modifier and the password of that user.
- ▶ The ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT lines activate an exit that notifies SMAPI of changes that are made to the user directory.
- ▶ If privacy of residual data is a concern on your system, use DISK_CLEANUP=YES.
- ▶ The ONLINE= IMMED line sets your changes to be made immediately.
- ▶ The RUNMODE= OPERATIONAL line sets directory changes to be committed. This line can be set to TESTING and the changes are not performed.

AUTHFOR control file

This file defines authorized administrators for your system in DirMaint. Several administrators can be defined in the AUTHFOR CONTROL file. The file is not included with the product and must be created manually. The file *must* be on the DIRMAINT 1DF disk. This file is also case-sensitive and *must* be in uppercase.

As shown in Figure 6-1 on page 161, a set of service VMs are granted full DirMaint authority through DirMaint specific privilege classes (ADGHOPS). These privilege classes represent distinct types of DirMaint roles and correspond to particular DirMaint commands and operations. When updating the AUTHFOR file, carefully consider the requirements of the VMs that are added and the scope of their authority.

```

ALL LNXADMIN * 140A ADGHOPS
ALL LNXADMIN * 150A ADGHOPS
ALL MAINT710 * 140A ADGHOPS
ALL MAINT710 * 150A ADGHOPS
ALL MAINT      * 140A ADGHOPS
ALL MAINT      * 150A ADGHOPS
ALL VSMGUARD  * 140A ADGHOPS
ALL VSMGUARD  * 150A ADGHOPS
ALL VSMWORK1  * 140A ADGHOPS
ALL VSMWORK1  * 150A ADGHOPS
ALL VSMWORK2  * 140A ADGHOPS
ALL VSMWORK2  * 150A ADGHOPS
ALL VSMWORK3  * 140A ADGHOPS
ALL VSMWORK3  * 150A ADGHOPS
ALL EDI       * 140A ADGHOPS
ALL EDI       * 150A ADGHOPS

```

Figure 6-1 Administrators who are authorized in DirMaint

The AUTHFOR CONTROL file specifies which VMs include the authority to modify the characteristics of other VMs on the system. This authority can be limited to specific target VMs or to specific attributes of a target VM. This authority is implemented by using DirMaint command sets. The format of this file is column-specific.

RPWLIST DATA

The RPWLIST DATA file is on the MAINT 2CC minidisk. When installing DIRMAINT, link and access this disk and copy this file to the 11F disk that is owned by DIRMAINT. It can be used to disable passwords that you deem to be trivial.

EXTENT CONTROL

The EXTENT CONTROL file defines disks (volumes) to DirMaint for minidisk allocation. It also contains system and device default values that are used during allocation operations. The following sections must be populated:

- ▶ Regions define the actual disks and their sizes to DirMaint. The **AUTOR** keyword can be used in user directory entries to take space from the regions. As a preferred practice, the region name and volume label always should be identical.
- ▶ Groups define pools of disks so that the **AUTOG** keyword can be used to take space from the pools, not from specific disks.

Note: For more information about DIRMAINT, see *Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283.

DirMaint-RACF Connector

RACF can coexist with the DirMaint product installed. However, to avoid dual maintenance of password processing (and other RACF functions), complete the following steps:

1. Use the DirMaint supplied sample file CONFIGRCSAMPDVH. You must copy this file to the 7VMDIR10 11F disk as CONFIGRCSAMPDVH.

Note: For more information about this file, see Chapter 3, “Tailoring the DIRMAINT Service Machine”, in *Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283. For this file to take effect, perform an IPL of DirMaint or run the **DIRM RLDDATA** command.

2. If RACF administration is centralized, you must give the DIRMAINT user ID RACF administrator SPECIAL authority. If RACF administration is decentralized, you must give the DIRMAINT user ID RACF group-SPECIAL authority.
3. If you want to record DirMaint activity in RACF SMF records, enable ESM_LOG_RECORDING_EXIT.
To do this, remove the comment from the item ESM_LOG_RECORDING_EXIT in the CONFIGRC DATADVH file. For this change to take effect, perform an IPL of DirMaint or run the **DIRM RLDDATA** command.
4. Authorize the DirMaint service machines DIRMAINT, DATAMOVE, and DIRMSAT to use the RACROUTE interface.

Note: For more information, see *Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283, and *z/VM: Security Server RACROUTE Macro Reference*, SC24-6324.

6.2.2 Delegating DirMaint authority

DirMaint access management include the following components:

- ▶ Command privilege classes
- ▶ AUTHFOR CONTROL and AUTHBY CONTROL files
- ▶ Exit routines

These mechanisms provide the management of administrators that are authorized to use DirMaint and the level of authority they have over their own user ID and the IDs of others.

In addition, DirMaint allows for commands to be issued under the authority of another user by using the **ASuser** prefix. Although this method is used when issuing commands to another system, it can also be used as a method for delegation of authority.

Command privilege classes

DirMaint uses a privilege class structure that is similar to that used by CP. Commands are grouped into classes based on the administrative function they perform, and users are then assigned to a class. A command can exist in one or more classes, and a user can also be assigned privileges over more than one class. The 140CMDS DATADVH and 150CMDS DATADVH files contain the mapping of commands into classes.

Custom classes can be created in the 1*0CMDS DATADVH files, which allows for a DirMaint administrator to create groupings of commands that suit the requirements of the installation. This process is described in “Defining a Custom Command Set” in the *Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283.

AUTHFOR CONTROL and AUTHBY CONTROL

The AUTHFOR CONTROL file on the DIRMAINT 1DF disk maps DirMaint administration users to the users they are allowed to administer and the command privileges they have over those users. An example of the AUTHFOR CONTROL file is shown in Example 6-2 on page 163.

Example 6-2 Example of the AUTHFOR CONTROL file

```
ALL LNXADMIN * 140A ADGHOPS
ALL LNXADMIN * 150A ADGHOPS
ALL LNXMAINT * 140A ADGHOPS
ALL LNXMAINT * 150A ADGHOPS
ALL MAINT * 140A ADGHOPS
ALL MAINT * 150A ADGHOPS
ALL MAINT710 * 140A ADGHOPS
ALL MAINT710 * 150A ADGHOPS
```

In this example, the users LNXADMIN, LNXMAINT, MAINT, and MAINT710 can run commands from both of the DirMaint command levels (140A and 150A) in the classes A, D, G, H, O, P, and S against all users on the system (the **ALL** keyword at the start of the records).

The AUTHFOR CONTROL file can be updated by running the DirMaint **AUTHFOR** command, or by running **DIRM SEND** to send a copy of the file, editing it directly, and running **DIRM FILE** to store it back to DirMaint. If the file is edited directly, the **DIRM RLDCODE** command must be run to refresh DirMaint with the update.

Note: For more information about AUTHFOR CONTROL, see *Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283.

Running commands by using ASuser

On the example system, Linux system administrators maintain Linux VM definitions by using DirMaint. Currently, the AUTHFOR CONTROL file grants access to these administrators over all the VMs on the system (by using the **ALL** keyword).

Suppose that you want to give these administrators access to operate only on the Linux VM users. By using AUTHFOR CONTROL, the only way this process can be done is to specifically list each Linux VM and the command privilege that applies for each administrator. This process introduces the following considerations:

- ▶ Each time a Linux is added, AUTHFOR CONTROL must be updated to add authority to the new guest for *all Linux administrators*.
- ▶ If an administrator joins or leaves the team, AUTHFOR CONTROL must be modified to add or remove entries for *all the Linux guests*.
- ▶ If the command privilege level that the Linux administrators should be assigned over Linux guests changes, *every* corresponding line in AUTHFOR CONTROL must be updated.

A method that can be used to implement a group membership approach is to define an administrator ID for each set of Linux guests. Individual administrators then use the **ASuser** prefix keyword on the **DIRM** command to issue their administrative commands to DirMaint under the authority of the group ID, as shown in the following example:

```
DIRM AS LNX1GRP FOR LNXS0030 REVIEW
```

This process works as a way to reduce the configuration effort because the only ID that appears in the AUTHFOR CONTROL file is the group administrator ID. However, consider the following drawbacks:

- ▶ The group administrator ID must be defined to RACF. It does not have to be defined to the CP directory.
- ▶ The **ASuser** prefix keyword forces a prompt for the administrator password whenever it is used.

- The password that must be provided when **ASuser** is used as the password of the group administrator ID.

Using BYuser

The **ASuser** prefix can be combined with BYuser to avoid having to know the password of the administrator ID. The use of BY with AS allows an administrator to override the password prompt that comes from using AS with a prompt for their own password, but the administrator must use their own ID as the option on the **BY** prefix. The DirMaint command then appear as shown in the following example:

```
DIRM AS LNX1GRP BY EDI FOR LNXICP1 REVIEW
```

A DirMaint administrator must be authorized to run commands by using BYuser because the password of the administrator who is running the command is used. Therefore, administrators on the system must be protected from other admins that are running commands on their behalf.

The authority is managed in a configuration file within DirMaint in a similar fashion to **FOR** by using the file AUTHBY CONTROL. The AUTHBY CONTROL file is maintained in a similar way to AUTHFOR CONTROL; that is, by using the DirMaint **AUTHBY** command or by directly editing the AUTHBY CONTROL file.

Note: Like AUTHFOR CONTROL, no supplied AUTHBY CONTROL file is available in a DirMaint installation. Use the **AUTHBY** command to create the first AUTHBY entry so that DirMaint creates the file on the correct disk. You can then use the SEND/FILE process to edit the file directly if you want.

To enable the **AUTHBY** command, the LNX1GRP user runs the following command:

```
DIRM AUTHBY EDI
```

If another administrator can access run commands on behalf of LNX1GRP, they also can run the following command:

```
DIRM FOR LNX1GRP AUTHBY EDI
```

Either of these commands result in a line being added to the AUTHBY CONTROL file, as shown in Example 6-3.

Example 6-3 Line added to AUTHBY CONTROL

```
LNX1GRP EDI
```

Suppressing the password prompt

DirMaint includes a configuration option ALLOW_ASUSER_NOPASS_FROM, which allows authorized administrators to run commands by using the **ASuser** prefix without being prompted for a password. This option is used only for the SMAPI worker servers; other DirMaint administrators should not be entered in this option. Therefore, it is not possible to suppress the password prompt for a DirMaint administrator by using the **ASuser** prefix.

Exit routines

In Chapter 9 “Exit routines” of *Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283, the exits are described that are available to modify the way DirMaint operates. Many of the exits that are available have an influence over command processing, which is useful because it allows more granular access control than what is provided by command privilege or administrator configuration.

Example exit usage: FOR authorization

The following sections describe an example of how a DirMaint exit can be used. The example implements group-based administration by using **ASuser** and **BYuser** by using a DirMaint exit instead of **AUTHFOR** and **AUTHBY**.

An easier way to achieve the requirement might be to use the DirMaint exit routine for the **FOR** command. Because the Linux administrators are using the **DIRM FOR** command to perform operations on the guests they manage, the use of the exit routine that is called when **FOR** commands are run is a good way to determine programmatically whether the command should be granted.

An example exit routine to achieve the effect is shown in Example 6-4.

Example 6-4 Sample DVHXFA EXEC for authority delegation

```
/* REXX */
/* DVHXFA EXEC */
/* DirMaint FOR Authorization Checking exit */
Address 'COMMAND'
Parse Upper Arg Asuser NodeID TgtID TgtNode CmdLv1 CmdSet Cmd
/* Read in the user file to find the group(s) */
'PIPE < USR2GRP DIRMFILE | ',
  'LOCATE /'TgtID'/ | ',
  'STEM Groups.'
/* Does the user belong to a group? Exit if not */
If Groups.0=0 Then Exit 30
/* For each group the user is a member of, see if the issuer is a member */
Do counter=1 to Groups.0
  Group=Word(Groups.counter,2)
  'PIPE < GRP2ADM DIRMFILE | ',
    'LOCATE /'Group'/ | ',
    'LOCATE /'Asuser'/ | ',
    'STEM Admin.'
/* If the stem is empty we try again; if not, we got one, we're done */
  If Admin.0=0 Then Iterate; Else Exit 0
End
Exit 30
```

The exec uses two files that are stored on a DirMaint disk, **USR2GRP DIRMFILE** and **GRP2ADM DIRMFILE** to provide group-based organization to DirMaint access control. The **USR2GRP DIRMFILE** file links z/VM user IDs to the group they belong to, and **GRP2ADM DIRMFILE** maps the administration groups to the DirMaint administrator users who are permitted to operate on them.

Records in **USR2GRP DIRMFILE** feature the following format:

Userid Group

One space character is sufficient between the user ID and the group.

Records in **GRP2ADM DIRMFILE** feature the following format:

Group Adminid [Adminid ...]

You can specify the group multiple times if needed to support many administrator IDs.

Implementing the DVHXFA exit

In this example, you implement the **DVHXFA EXEC** exit to ensure it works as expected. Complete the following steps:

1. Install the files onto the DIRMAINT 11F disk by running **DIRM FILE** (specifying the destination file mode of D to make sure that they went to 11F instead of the 191):

```
DIRM FILE DVHXFA EXEC A = = D
DIRM FILE USR2GRP DIRMFILE A = = D
DIRM FILE GRP2ADM DIRMFILE A = = D
```
2. Add the correct entry to the CONFIG* DATADVH file set to activate the exit:

```
FOR_AUTHORIZATION_CHECKING_EXIT= DVHXFA EXEC
```
3. You have a CONFIGAA DATADVH file that contains all the local changes, so add the line there by running **DIRM SEND**, receive, edit, and the run **DIRM FILE**.
4. Activate the altered configuration by running **DIRM RLDDData** to put the exit into service.
5. Remove the authorization for the Linux administrators from AUTHFOR CONTROL by using a bulk update by using the SEND-receive-edit-FILE-RLDD method.
6. Ensure that the administrators can do what they need. In this example, the administrator EDI was permitted to administer only the groups CMSGRP and LNX1GRP. LNX1GRP contained the user LNXS0030, but another user LNXS0038 was in a different group. The CMS user USERBOB was part of CMSGRP.

Example 6-5 shows the results when administrator EDI attempted to work on system user IDs.

Example 6-5 Run DirMaint commands when the DVHXFA exit is in place

dirm for userbob review

```
DVHXMT1191I Your REVIEW request has been sent for processing to DIRMAINT
DVHXMT1191I at ITS0ZVM1.
Ready; T=0.01/0.01 17:18:40
DVHREQ2288I Your REVIEW request for USERBOB at * has been accepted.
RDR FILE 0237 SENT FROM DIRMAINT PUN WAS 6593 RECS 0028 CPY 001 A NOHOLD NOKEEP
DVHREQ2289I Your REVIEW request for USERBOB at * has completed; with
DVHREQ2289I RC = 0.
```

dirm for lnxS0030 review

```
DVHXMT1191I Your REVIEW request has been sent for processing to DIRMAINT
DVHXMT1191I at ITS0ZVM1.
Ready; T=0.01/0.01 17:19:11
DVHREQ2288I Your REVIEW request for LNXS0030 at * has been accepted.
RDR FILE 0241 SENT FROM DIRMAINT PUN WAS 6597 RECS 0045 CPY 001 A NOHOLD NOKEEP
DVHREQ2289I Your REVIEW request for LNXS0030 at * has completed; with RC
DVHREQ2289I = 0.
```

dirm for lnxS0038 review

```
DVHXMT1191I Your REVIEW request has been sent for processing to DIRMAINT
DVHXMT1191I at ITS0ZVM1.
DVHREQ2287E User EDI at ITS0ZVM1 is not authorized to act for LNXS0038
DVHREQ2287E at *; your request is ignored.
```

dirm for maint710 review

```
DVHXMT1191I Your REVIEW request has been sent for processing to DIRMAINT
DVHXMT1191I at ITS0ZVM1.
DVHREQ2287E User EDI at ITS0ZVM1 is not authorized to act for MAINT710
DVHREQ2287E at *; your request is ignored.
```

6.3 Systems Management API

The z/VM SMAPI is the access point for any external tool to manage the z/VM running on IBM LinuxONE platform. It supports management of lifecycle and configuration of various platform resources, such as Guest, CPU, memory, virtual switches, storage, and more.

Some IBM products use the SMAPI to perform various tasks on the z/VM system (see Figure 6-2). Therefore, it is necessary to make sure that SMAPI is configured and running before you configure any cloud piece that interacts with z/VM. The exact configuration steps for SMAPI might differ from the following section based on the version and release level of z/VM.

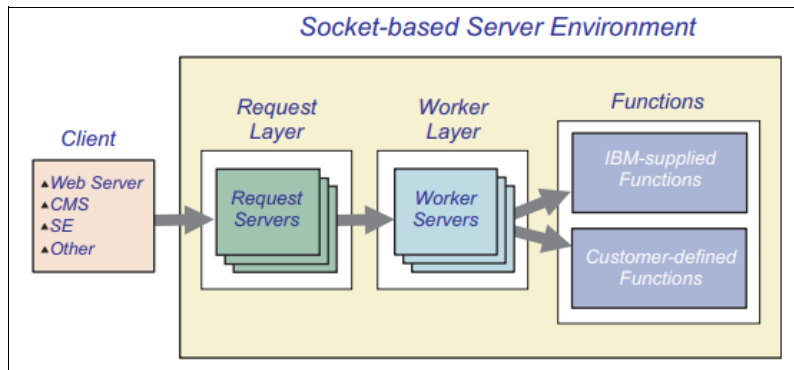


Figure 6-2 SMAPI socket-based servers work together

Note: Because this IBM Redbooks publication references the current version of CMA Newton (6.4), we must use the SMAPI 6.4 to be compatible and it is included with CMA 6.4.

6.3.1 SFS

The SMAPI request servers and worker servers use Shared File System (SFS) directories to access configuration files and other data. SMAPI uses the standard file pool VMSYS and VMPSFS to keep their files. The VSMWORK1 user ID is the owner of some of the SFS directories that have control files, logs, and so on.

Security aspects with SFS directories

The SFS directories are defined on SFS file pools. The authorization and ownership for the SFS directories are done by using enroll SFS commands. You can set AUDIT parameter on DSMPARMS file for auditing purpose.

Note: For more information about managing and auditing the VMSYS or VMPSFS file pools, see *z/VM: CMS File Pool Planning, Administration, and Operation*, SC24-6261.

All commands that are shown in this chapter regarding SFS **ENROLL** and **GRANT** are performed automatically during z/VM installation. They are shown here for verification and testing purposes.

Also, if you are adding a worker or request server, you can use the appropriate commands from these lists as a guide for enrolling your new server in the correct file pool and then grant SFS authorizations (see Example 6-6 on page 168).

Example 6-6 SFS ENROLL command

```
ENROLL USER VSMWORK1 VMSYS: (BLOCKS 6000 STORGROUP 2
ENROLL USER VSMWORK2 VMSYS:
ENROLL USER VSMWORK3 VMSYS:
ENROLL USER VSMREQIN VMSYS:
ENROLL USER VSMREQIU VMSYS:
ENROLL USER VSMGUARD VMPSFS: (BLOCKS 1000 STORGROUP 2
ENROLL USER VSMGUARD VMSYS:
ENROLL USER VSMREQI6 VMSYS:
ENROLL USER VSMEVSRV VMSYS:
ENROLL USER DTCSMAPI VMSYS:
ENROLL USER OPERATNS VMSYS:
ENROLL USER PERSMAPI VMSYS: (BLOCKS 24000 STORGROUP 2
ENROLL USER OPNCLOUD VMSYS:
```

If you do not grant access to the specific directory, you cannot access it. Example 6-7 shows SFS **GRANT** commands that are automatically performed during z/VM installation.

Example 6-7 SFS GRANT command

```
GRANT AUTHORITY VMSYS:VSMWORK1. TO OPNCLOUD (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1.DATA TO OPNCLOUD (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1. TO MAINT (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1.DATA TO MAINT (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1. TO VSMGUARD (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1.DATA TO VSMGUARD (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1.STATUS TO VSMGUARD (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1.STATUS TO VSMWORK2 (WRITE NEWWRITE
GRANT AUTHORITY VMSYS:VSMWORK1.STATUS TO VSMWORK3 (WRITE NEWWRITE
GRANT AUTHORITY * * VMSYS:VSMWORK1. TO VSMGUARD (READ
GRANT AUTHORITY VMSYS:VSMWORK1. TO PERSMAPI (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DIRMAINT (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DIRMSAT (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DIRMSAT2 (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DIRMSAT3 (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DIRMSAT4 (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DATAMOVE (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DATAMOV2 (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DATAMOV3 (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO DATAMOV4 (READ NEWREAD
GRANT AUTHORITY VMPSFS:VSMGUARD. TO AUTOLOG1 (WRITE NEWWRITE
GRANT AUTHORITY VMPSFS:VSMGUARD. TO AUTOLOG2 (WRITE NEWWRITE
```

Note: For more information about SMAPI, see *The Virtualization Cookbook for IBM z Systems@Volume 1: IBM z/VM 6.3*, SG24-8147, *Systems Management Application Programming (6.4)*, SC24-6327, and *Enabling z/VM for OpenStack (Newton)*, SC24-6253.

6.3.2 Other SMAPI user IDs

The interaction with SMAPI occurs through a client/server architecture and SMAPI includes the following types of servers:

- ▶ Request
- ▶ Worker

Request servers

A listening request server receives a connection request from a client, establishes a connection, and then accepts requests from that client. The following request servers are available:

- ▶ VSMREQIN
- ▶ VSMREQI6
- ▶ VSMREQIU
- ▶ VSMEVSRV

Worker servers

The worker servers process API function requests. The following worker servers are defined in the default installation:

- ▶ VSMWORK1
- ▶ VSMWORK2
- ▶ VSMWORK3

A fourth worker server, VSMGUARD, is also defined. VSMGUARD is a “guard” server that helps provide better resiliency and error recovery. For more information, see 6.3.3, “VSMGUARD” on page 169.

Other servers

The following servers also are available:

- ▶ The LOHCOST server is used for caching the system directory contents that are required to satisfy the various query APIs. It is also used to store and retrieve data that is used by the metadata APIs.
- ▶ The DTCSMAPI server is used by several of the SMAPI servers for communication and workload balancing.
- ▶ The PERSMAPI server is used for performance monitoring.
- ▶ The VSMEVSRV server is used to listen for and then propagate VMEVENT and directory updates.
- ▶ The OPERATNS server is used to collect, format, and distribute ABEND memory dumps.

6.3.3 VSMGUARD

The VSMGUARD worker server grants authority to all the other SMAPI servers that are configured to access the SMAPI file space. Therefore, VSMGUARD must be made an administrator of the VMSYS: file pool. This process is done by adding VSMGUARD to the list of users that are authorized for ADMIN authority.

Note: In the default environment, this process is done by updating the VMSERVS DMSPARMS file on the VMSERVS 191 disk.

VSMGUARD has an important role in the SMAPI environment. When you must recycle all SMAPI user IDs, you do it by recycling VSMGUARD (**force** and **xautolog**) and it recycles all the other SMAPI user IDs, saves the SMAPI segment, and defines the vSwitches that are listed in the configuration file.

Note: For more information about SMAPI, see *The Systems Management Application Programming for 6.4*, SC24-6327.

6.3.4 SMAPI controls

Because xCAT uses SMAPI to interact with the system to enable xCAT on z/VM, you must configure the following SMAPI files:

- **DMSSISVR NAMES**

DMSSISVR NAMES is a CMS NAMES file that defines each specific request and worker servers in the z/VM environment. This file is on the MAINT the new 1193 MDisk that is included with the currently CMA 6.4 package.

Modify the DMSSISVR NAMES file and uncomment the directory manager and the memory dump handler definitions. You can modify it manually or run the VMSES/E **LOCALMOD** command to change this file.

Note: The use of VM/SES helps preserve your configuration changes if IBM makes service updates or future release update in this file.

- **DMSSICNF COPY**

The DMSSICNF COPY file contains several global attributes that can be modified to better fit your installation and networking configurations, such as IP addresses, gateway, netmask, domain name, and vSwitches. This file is the heart of SMAPI and xCAT because it is used to control the definition of the servers when it is initialized. This file is on MAINT new 1193 MDisk and is included with the current CMA 6.4 package.

To make the DMSSISVR and DMSSICNF files changes available to SMAPI and the OPNCLOUD server user IDs, run the following VM/SES commands after the files are updated:

```
SERVICE CMS BUILD  
PUT2PROD
```

6.3.5 Security aspects of SMAPI

An ESM controls who can have access, and what kind of access they can have, to specific resources. If an ESM is implemented at your installation, SMAPI must be given the appropriate access to the disks, SFS directories, and resources you want it to manage. In this example installation, use RACF as the ESM.

In addition to the security aspects that you have by using SFS, you have other authority files on SMAPI that list who is authorized to run commands on SMAPI. Make sure your installation grants access to authorized people only.

VSMWORK1 AUTHLIST

Authenticated users must be authorized to issue API requests. A server authorization file is used for this purpose. The authorization file contains entries that authorize authenticated users to perform specific functions for specific virtual images (target users) or lists of virtual images. The authorization can be granted per requesting VM, per target, or per function. This file is in the VMSYS file pool, under the VSMWORK1 SFS directory and, during the installation, VSMGUARD is granted access to it, as shown in Example 6-8.

Example 6-8 VSMGUARD access to the VSMWORK1 directory

```
grant authority vmsys:vsmwork1. to vsmguard (write newwrite  
grant authority vmsys:vsmwork1.data to vsmguard (write newwrite  
grant authority * * vmsys:vsmwork1. to vsmguard (read
```

Using SMAPI with RACF

RACF for z/VM can be used to enhance the security and integrity of your system in the following ways:

- ▶ Helping you to implement the company's security policy
- ▶ Identifying and authenticating each user
- ▶ Controlling each user's access to sensitive data
- ▶ Logging and reporting events that are relevant to the system's security

Enabling RACROUTE

Enable the SMAPI service machines VSMREQI6, VSMREQIN, VSMREQIU, VSMEVSRV, DTCSMAPI, VSMWORK1, VSMWORK2, and VSMWORK3 to use RACROUTE services, as shown in Example 6-9.

Example 6-9 RACF RACROUTE definitions for SMAPI user IDs

```
RAC SETROPTS CLASSACT(FACILITY)
RAC SETROPTS RACLIST(FACILITY)
RAC RDEFINE FACILITY ICHCONN UACC(NONE)
RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMREQI6) ACCESS(UPDATE)
RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMREQIN) ACCESS(UPDATE)
RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMREQIU) ACCESS(UPDATE)
RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMEVSRV) ACCESS(UPDATE)
RAC PERMIT ICHCONN CLASS(FACILITY) ID(DTCSMAPI) ACCESS(UPDATE)
RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMWORK1) ACCESS(UPDATE)
RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMWORK2) ACCESS(UPDATE)
RAC PERMIT ICHCONN CLASS(FACILITY) ID(VSMWORK3) ACCESS(UPDATE)
RAC SETROPTS RACLIST(FACILITY) REFRESH
```

The directory entry for the SMAPI service machines that use this capability must all contain the following statement:

```
IUCV ANY PRIORITY MSGLIMIT 255
```

An MSGLIMIT value of 255 is initially suggested. It can be adjusted as needed.

Making SMAPI user IDs exempt for some RACF checking

The SMAPI service machines (DTCSMAPI, VSMWORK1, VSMWORK2, and VSMWORK3) should be made exempt from access checking. Even if access checking is not active on your system, make the SMAPI service machines exempt from access checking for the **FOR** (privilege class C), and **LINK** commands, as shown in Example 6-10.

Example 6-10 Make SMAPI user IDs exempt for FOR and LINK commands

```
RAC SETROPTS CLASSACT(VMXEVENT)
RAC RDEFINE VMXEVENT USERSEL.DTCSMAPI
RAC RALTER VMXEVENT USERSEL.DTCSMAPI ADDMEM(FOR.C/NOCTL)
RAC RALTER VMXEVENT USERSEL.DTCSMAPI ADDMEM(LINK/NOCTL)
RAC SETEVENT REFRESH USERSEL.DTCSMAPI
RAC RDEFINE VMXEVENT USERSEL.VSMWORK1
RAC RALTER VMXEVENT USERSEL.VSMWORK1 ADDMEM(FOR.C/NOCTL)
RAC RALTER VMXEVENT USERSEL.VSMWORK1 ADDMEM(LINK/NOCTL)
RAC SETEVENT REFRESH USERSEL.VSMWORK1
RAC RDEFINE VMXEVENT USERSEL.VSMWORK2
RAC RALTER VMXEVENT USERSEL.VSMWORK2 ADDMEM(FOR.C/NOCTL)
RAC RALTER VMXEVENT USERSEL.VSMWORK2 ADDMEM(LINK/NOCTL)
RAC SETEVENT REFRESH USERSEL.VSMWORK2
```

```
RAC RDEFINE VMXEVENT USERSEL.VSMWORK3
RAC RALTER VMXEVENT USERSEL.VSMWORK3 ADDMEM(FOR.C/NOCTL)
RAC RALTER VMXEVENT USERSEL.VSMWORK3 ADDMEM(LINK/NOCTL)
RAC SETEVENT REFRESH USERSEL.VSMWORK3
```

Making the OPNCLOUD user ID exempt from transfer command access validation

If your system is using xCAT integrated to CMA, the OPNCLOUD service machine must be made exempt from spool checking so that it can transfer files to various virtual machines. Do this by running the commands that are shown in Example 6-11.

Example 6-11 Exempt the ZHCP user ID for access command validation

```
RAC SETROPTS CLASSACT(VMXEVENT)
RAC RDEFINE VMXEVENT USERSEL.OPNCLOUD
RAC RALTER VMXEVENT USERSEL.OPNCLOUD ADDMEM(TRANSFER.G/NOCTL)
RAC SETEVENT REFRESH USERSEL.OPNCLOUD
```

Enabling SMAPI to access DIAGNOSE X'88'

You must enable the SMAPI service machines for DIAGNOSE X'88' access. If RACF is used to control DIAGNOSE X'88' access, enable DIAGNOSE X'88' access for SMAPI by completing the following steps:

1. Enable RACF/VM profile protection for DIAGNOSE X'88', as shown in Example 6-12.

Example 6-12 Create a profile DIAG88 in VMCMD class

```
RAC RDEFINE VMCMD DIAG088 UACC(NONE)
RAC SETROPTS CLASSACT(VMCMD)
```

Note: Each SMAPI server has the OPTION DIAG88 statement in its directory entry. If you do not enable RACF protection, the checking defaults to the CP directory OPTION DIAG88 entry, which tells CP that the server is authorized to use DIAGNOSE code X'88'.

2. Give the SMAPI server permission to perform password validation (which uses DIAGNOSE X'88' subcode 8), as shown in Example 6-13, Example 6-14, and Example 6-15 on page 173.

Example 6-13 Give authority to the requester servers

```
RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMREQIN) ACCESS(READ)
RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMREQI6) ACCESS(READ)
RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMREQIU) ACCESS(READ)
RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMEVSRV) ACCESS(READ)
```

Example 6-14 Give authority to the worker servers

```
RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMGUARD) ACCESS(READ)
RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMWORK1) ACCESS(READ)
RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMWORK2) ACCESS(READ)
RAC PERMIT DIAG088 CLASS(VMCMD) ID(VSMWORK3) ACCESS(READ)
```

Example 6-15 Give authority to these SMAPI user IDs

```
RAC PERMIT DIAG088 CLASS(VMCMD) ID(LOHCOST) ACCESS(READ)
RAC PERMIT DIAG088 CLASS(VMCMD) ID(DTCSMAPI) ACCESS(READ)
RAC PERMIT DIAG088 CLASS(VMCMD) ID(PERSMAPI) ACCESS(READ)
RAC PERMIT DIAG088 CLASS(VMCMD) ID(OPERATNS) ACCESS(READ)
```

Attention: After creating DIAG088, you must issue the permit command to DIRMAINT.

Enabling SMAPI to access needed resources

You must enable the SMAPI service machine for minidisk, reader, and VMBATCH access, as shown in Example 6-16, Example 6-17, Example 6-18, and Example 6-19.

Example 6-16 For minidisk access: RACF uses to control minidisk access

```
RAC PERMIT MAINT710.5E5 CLASS(VMMDISK) ID(VSMWORK1) ACCESS(READ)
RAC PERMIT MAINT710.51D CLASS(VMMDISK) ID(VSMWORK1) ACCESS(READ)
RAC PERMIT PMAINT.551 CLASS(VMMDISK) ID(VSMGUARD) ACCESS(READ)
```

Example 6-17 Allow VSMWORK1 minidisk authority

```
RAC PERMIT PMAINT.CF0 CLASS(VMMDISK) ACC(CONTROL) ID(VSMWORK1)
RAC PERMIT MAINT.CF1 CLASS(VMMDISK) ACC(CONTROL) ID(VSMWORK1)
```

Example 6-18 Allow SMAPI worker servers to read the TCPMAINT 198 disk

```
RAC PERMIT TCPMAINT.198 CLASS(VMMDISK) ACC(READ) ID(VSMGUARD)
RAC PERMIT TCPMAINT.198 CLASS(VMMDISK) ACC(READ) ID(VSMWORK1)
RAC PERMIT TCPMAINT.198 CLASS(VMMDISK) ACC(READ) ID(VSMWORK2)
RAC PERMIT TCPMAINT.198 CLASS(VMMDISK) ACC(READ) ID(VSMWORK3)
```

Example 6-19 Enable reader access to DTCSMAPI for MAINT and TCPMAINT user IDs

```
RAC PERMIT MAINT CLASS(VMRDR) ID(DTCSMAPI) ACCESS(UPDATE)
RAC PERMIT TCPMAINT CLASS(VMRDR) ID(DTCSMAPI) ACCESS(UPDATE)
```

VMBATCH access

Permit the SMAPI servers CONTROL access to a generic VMBATCH, or to an existing discrete VMBATCH profile to use the SMAPI services, as shown in Example 6-20, Example 6-21, and Example 6-22 on page 174.

Example 6-20 RACF commands to define VMBATCH

```
rac setropts generic(vmbatch) gencmd(vmbatch)
rac rdefine vmbatch ** uacc(none)
rac permit ** class(vmbatch) id(ftpserve vmnfs dirmsat dirmsat2) acc(control)
rac setropts classact(vmbatch vmmdisk vmcmd vmlan surrogat)
```

Example 6-21 Give CONTROL access if you have an existing generic VMBATCH profile

```
RAC PERMIT ** CLASS(VMBATCH) ID(VSMWORK1) ACCESS(CONTROL)
RAC PERMIT ** CLASS(VMBATCH) ID(VSMWORK2) ACCESS(CONTROL)
RAC PERMIT ** CLASS(VMBATCH) ID(VSMWORK3) ACCESS(CONTROL)
RAC PERMIT ** CLASS(VMBATCH) ID(DTCSMAPI) ACCESS(CONTROL)
```

Example 6-22 Give CONTROL access if you have an existing generic VMBATCH profile:

```
RAC PERMIT CLASS(VMBATCH) ID(VSMWORK1) ACCESS(CONTROL)
RAC PERMIT CLASS(VMBATCH) ID(VSMWORK2) ACCESS(CONTROL)
RAC PERMIT CLASS(VMBATCH) ID(VSMWORK3) ACCESS(CONTROL)
RAC PERMIT CLASS(VMBATCH) ID(DTCSMAPI) ACCESS(CONTROL)
```

Although all of the items that are described here are important, they are not enough without validating them. Auditing SMAPI requests ensures that the security policies that are applied are being followed and are correctly assigned.

SMAPI ESM authorization support

SMAPI provides the following Enterprise Security Manager (ESM) interaction:

- ▶ When an ESM is present, programs can use the ESM for all SMAPI authorization decisions at the same granularity that is used with the SMAPI existing authorization mechanism. The ESM logs (or does not log) the decision that is based on its active policy, without SMAPI knowledge or intervention.
- ▶ When an ESM defers its authorization decision to SMAPI, one of the following actions is taken based on a configuration option:
 - The SMAPI authorization decision uses the existing authorization process. SMAPI calls the ESM to log the decision in the ESM-managed security log. SMAPI has no knowledge if the ESM audit logging is enabled or disabled.
 - SMAPI treats the request as unauthorized

6.4 z/VM Cloud Manager Appliance

z/VM Cloud Manager Appliance (CMA) allows the use of OpenStack to deploy Linux guests on z/VM, and for the integration of z/VM into larger environments. The current CMA version is on OpenStack Newton and is supported as a z/VM component without extra license requirements.

CMA manages only z/VM platforms and it does not deploy guests onto non-z/VM platforms. The CMA changes provide several different options for using CMA: as stand-alone cloud or integrated with another OpenStack environment.

This section refers to the current version of z/VM CMA Newton.

Note: Consider the following points:

- ▶ The support for CMA Newton version on z/VM v7.1 is a transitional offer until a strategic long-term solution to replace the CMA becomes available.
- ▶ CMA Newton is available for z/VM v7.1 for the specific use case with IBM Cloud Private only.
- ▶ To get the CMA Newton package, you must contact your IBM representative or IBM account team.

Figure 6-3 shows an overall view of CMA Architecture for z/VM.

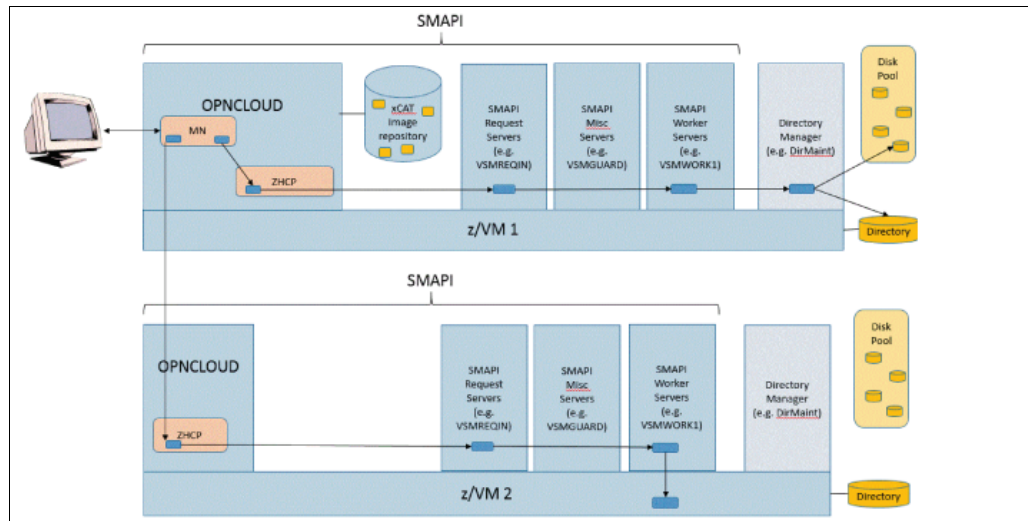


Figure 6-3 z/VM CMA architecture

6.4.1 Basic requirements and configuration options

Each z/VM logical partition (LPAR) requires a CMA. The extreme Cloud Administration Toolkit (xCAT) must always be active within the CMA. The OpenStack services are optional, and depend on the intended use.

CMA can be configured in different ways, based on the needed function or integration requirements. z/VM currently supports the following system roles, which control the set of services running inside the OPNCLOUD virtual machine:

- Controller node

Runs cloud controller services (such as the glance image services) in addition to all services that are listed under the compute role. z/VM also runs the xCAT MN and ZHCP services to allow the controller to manage OpenStack z/VM hosts.

- Compute node

Runs compute services (nova-compute service), networking services (neutron-zvm-agent service), and telemetry services (ceilometer-polling) for the z/VM hypervisor. z/VM also runs the ZHCP service to allow a remote xCAT MN service to manage the host.

- Compute MN

Runs compute, networking, and telemetry services (listed under the compute role) for the z/VM hypervisor. It also runs the xCAT MN and ZHCP services. This type of node is used in an environment where OpenStack controller services are run on a non-CMA node (for example, on other platforms). The xCAT MN and ZHCP services allow a controller to manage the z/VM host without requiring cloud controller services to be running on the host.

- Managed Node (MN)

Runs the xCAT MN and ZHCP services. This role is useful when all OpenStack services are running in non-CMA nodes or when you want to use xCAT and not OpenStack.

- ▶ ZHCP

Runs only the ZHCP service. This role is useful when all OpenStack services are running in non-CMA nodes or when you want to use xCAT and not OpenStack. Another z/VM host must run an xCAT MN service to manage the host through the ZHCP service.

Note: xCAT MN and ZHCP servers run within the same virtual machine, which is called the OPNCLOUD.

z/VM currently supports OpenStack Newton, which includes the following items:

- ▶ Support for the Newton release of OpenStack.
- ▶ Integration of the xCAT function into the z/VM CMA, which allows running a fully functional z/VM OpenStack solution in a single virtual server so that separate ZHCP servers are not required.
- ▶ Support for provisioning Red Hat RHEL 7 and SUSE SLES 12 servers.

For more information about z/VM CMA for OpenStack Newton, see *Enabling z/VM for OpenStack (Support for OpenStack Newton Release)*, SC24-6251, and *Systems Management Application Programming (6.4)*, SC24-6327.

Note: To get CMA Newton, contact your IBM representative or IBM account team.

6.4.2 OpenStack and xCAT Service Deployment Patterns

In this section, we provide an overview of the z/VM systems management architecture, OpenStack architecture, the CMA, which are environments that are available for using OpenStack with z/VM. We also include some examples for choosing the correct environment for your installation.

An OpenStack solution is free to run its components wherever it wants; its options range from running all components on z/VM, to running some on z/VM and others elsewhere, to running all components on other platforms. The solution is also free to source its components wherever it wants, by using z/VM's OpenStack enablement components or not using the components.

6.4.3 z/VM System Management Architecture

z/VM includes a set of servers that provide local system management APIs. These servers consist of request servers that accept local connections, receive the data, and then call one of a set of worker servers to process the request. These servers are known collectively as SMAPI. The worker servers can interact with the z/VM hypervisor (CP) or with a directory manager. A directory manager is required for this environment.

Since z/VM version 6.3, more functions are provided by integrated xCAT services. xCAT is an open source scalable distributed computing management and provisioning tool that provides a unified interface for hardware control, discovery, and deployment, including remote access to the SMAPI APIs. It can be used for the deployment and administration of Linux servers that OpenStack wants to manipulate. The z/VM drivers in the OpenStack services communicate with xCAT services by using REST APIs to manage the virtual servers.

xCAT is composed of two main services: the xCAT management node (xCAT MN) and ZHCP. The xCAT MN server and the ZHCP server run within the same virtual machine, which is called the OPNCLOUD virtual machine.

The xCAT MN coordinates creating, deleting, and updating virtual servers. The management node uses a z/VM hardware control point (ZHCP) to communicate with SMAPI to implement changes on a z/VM host. Only one instance of the xCAT MN is necessary to support multiple z/VM hosts. Each z/VM host runs one instance of ZHCP. xCAT MN supports both a GUI for human interaction and REST APIs for use by programs (for example, OpenStack).

Next, we describe two examples of how you can use this environment.

CMA with z/VM in controller node

You configure the CMA on one z/VM system in the cloud (for example, z/VM 1) to run in the “controller” role so it sets up the OPNCLOUD virtual server to run the OpenStack cloud controller, OpenStack compute (managing the system on which it is running; for example, z/VM 1), xCAT MN, and ZHCP services.

You configure the CMA on all other z/VM systems (for example, z/VM 2) in the cloud to run in the “compute” role so they each set up one OPNCLOUD virtual server to run the OpenStack compute and ZHCP services that manage the system on which those services are running.

Because the xCAT MN service running in z/VM 1 manages all z/VM systems in the cloud through the ZHCP service running on each z/VM system, you do not run the xCAT MN service on any other systems in the cloud (in this case, on z/VM 2).

Each CMA installs the OpenStack code and configures its z/VM drivers automatically because all CMAs here are configured to run OpenStack compute services. The virtualization manager calls OpenStack cloud controller APIs when it must interact with the virtual servers OpenStack deploys on z/VM; these APIs are served by the OPNCLOUD virtual server on z/VM 1.

For more information, see 6.5, “CMA Controller node” on page 178.

Services and virtual machines that run when a virtualization manager uses z/VM’s CMA as a cloud controller are shown in Figure 6-4. It also shows variations with one and two z/VM systems.

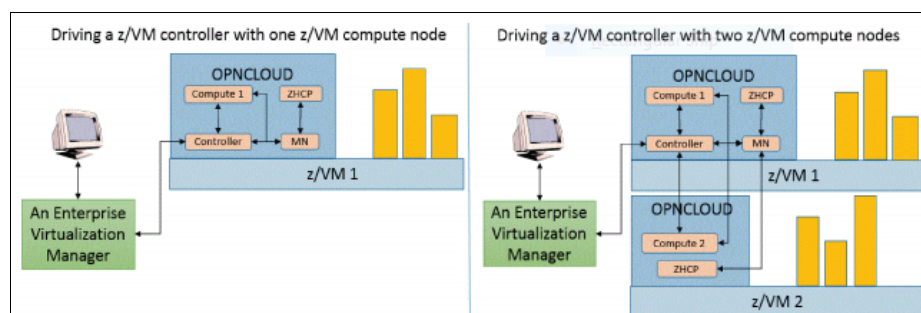


Figure 6-4 z/VM CMA in controller node

CMA with z/VM in an entry level cloud

You configure the CMA on one z/VM system in the cloud (for example, z/VM 1) to run in the “controller” role, so it sets up the OPNCLOUD virtual server to run the OpenStack cloud controller, OpenStack compute, xCAT MN, and ZHCP services.

You configure the CMA on all other z/VM systems (for example, z/VM 2) in the cloud to run in the “compute” role, so they each set up one OPNCLOUD virtual server to run the OpenStack compute and ZHCP services that manage the system on which those services are running. Because the xCAT MN service running in z/VM 1 manages all z/VM systems in the cloud through the ZHCP service running on each z/VM system, you do not run the xCAT MN service on any other systems in the cloud (in this case, on z/VM 2).

Each CMA installs the OpenStack code and configures its z/VM drivers automatically because all CMAs here are configured to run OpenStack compute services.

Figure 6-5 shows services and virtual machines that run when you use z/VM’s CMA as an entry level cloud, which you might do in preparation for adopting an OpenStack solution, for evaluation purposes, or to run an entry level z/VM-only cloud. It shows variations with one and two z/VM systems.

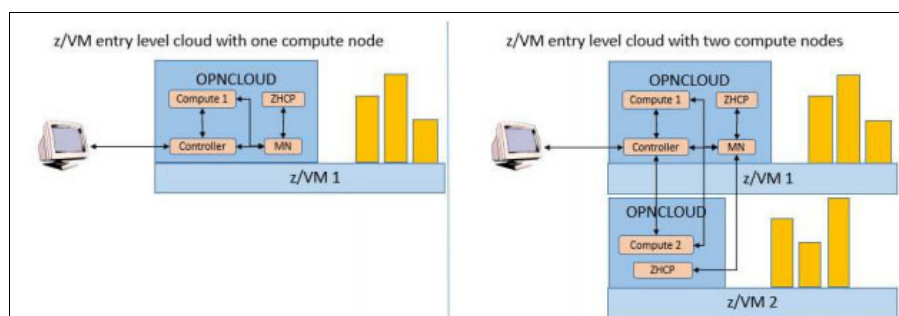


Figure 6-5 z/VM entry level cloud with compute node.

For more information about z/VM CMA for OpenStack Newton and server development patterns, see *Enabling z/VM for OpenStack (Support for OpenStack Newton Release)*, SC24-6251, and *Systems Management Application Programming (6.4)*, SC24-6327.

6.5 CMA Controller node

When the CMA is in controller node mode, it can be used in the following ways:

- ▶ A stand-alone OpenStack environment
CMA acts as an xCAT MN and controller and as the compute node and ZHCP for the z/VM LPAR on which it is running.
- ▶ Multi-region OpenStack environment
Other z/VM LPAR CMAs are defined as compute nodes and run the xCAT ZHCP function, and are controlled by the controller CMA.
- ▶ Integrated with other z/VM LPARs
Other z/VM LPAR CMAs are defined as compute nodes and run the xCAT ZHCP function, and are controlled by the controller CMA.

Figure 6-6 shows the CMA for z/VM controller and compute nodes.

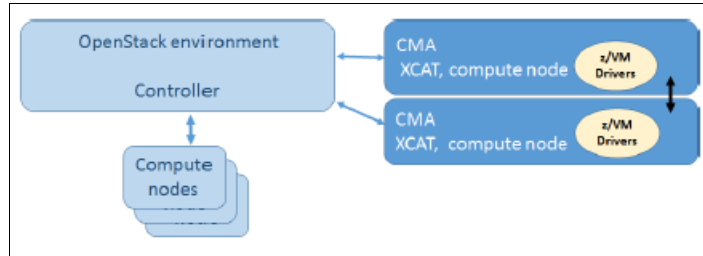


Figure 6-6 CMA for z/VM controller and compute nodes

The following files are used to configure the CMA Newton (6.4) as the Controller node:

- ▶ DMSSICNF COPY
- ▶ DMSSICMO COPY

6.5.1 DMSSICNF COPY for the controller node

The DMSSICNF COPY file contains several global attributes that can be modified to better fit your installation. In this case, it is defined on the RDBKZVM1 LPAR.

Note: When modifying this file, note that it is case-sensitive. When editing it, always run the **case mixed** command.

Example 6-23 shows a DMSSICNF COPY.

Example 6-23 DMSSICNF COPY example of using CMA

```

/*****
/* XCAT server defaults
/*****
XCAT_User      = "OPNCLOUD"           /* xCAT z/VM user ID
XCAT_Addr      = "10.10.10.10"        /* XCAT IP Address
XCAT_Host      = "opnccloud"          /* xCAT hostname
XCAT_Domain    = ".cpolab.ibm.com"    /* Main name
XCAT_vswitch   = "XCATVSW1"          /* xCAT Vswitch name
XCAT_OSAdev    = "NONE"              /* OSA address for xCAT
XCAT_zvmssid   = "RDBKZVM1"          /* xCAT z/VM system id
XCAT_notify    = "OPERATOR"          /* Notify when xCAT started
XCAT_gateway   = "NONE"              /* Network gateway IP addr.
XCAT_netmask   = "255.255.255.0"     /* Default network mask
XCAT_vlan      = "NONE"

XCAT_MN_Addr   = "9.76.61.215" /* xCAT mgmt node IP address */
XCAT_MN_vswitch = "XCATVSW2"      /* xCAT MN Vswitch name
XCAT_MN_OSAdev  = "1B14"          /* OSA address for xCAT MN
XCAT_MN_gateway = "9.76.61.1"     /* Network gateway IP addr.
XCAT_MN_Mask    = "255.255.255.0" /* Netmask for xCAT MN
XCAT_MN_vlan    = "2230"

XCAT_MN_admin  = "mnadmin"         /* MN administrator userid
XCAT_MN_pw     = "yourppw"         /* MN admin password
/* (if NOLOG, userid cannot
/* ssh into XCAT MN)

```

```

/*****/
/* ZHCP server defaults */
/*****/
ZHCP_User      = "OPNCLOUD"           /* zhcp z/VM user ID      */
ZHCP_Addr      = "NONE"               /* zhcp IP ADDRESS       */
ZHCP_Host      = "zhcpzvm1"           /* zhcp hostname         */
ZHCP_Domain    = ".cpolab.ibm.com"    /* Domain name           */
ZHCP_gateway   = "NONE"               /* Network gateway IP addr.*/
ZHCP_netmask   = "NONE"               /* Default network mask  */
ZHCP_vswitch   = "NONE"               /* zhcp Vswitch name     */
ZHCP_OSAdev    = "NONE"               /* OSA address for zhcp  */
ZHCP_vlan     = "NONE"
/*****/

```

The following configuration options are available:

XCAT_Domain	OpenStack controller domain name
XCAT_OSAdev	XCATVSW1 (for xCAT internal network) real OSA device address
XCAT_zvmsysid	z/VM SYSID where OpenStack controller runs (lowercase)
XCAT_MN_Addr	OpenStack controller (xCAT Management Node) IP address
XCAT_MN_OSAdev	XCATVSW2 (for management network) real OSA device address
XCAT_MN_gateway	OpenStack controller default gateway
XCAT_MN_netmask	OpenStack controller default netmask
XCAT_MN_pw	xCAT Management Node (mnadmin) default password, which should be changed during setup
ZHCP_Host	ZHCP host name
ZHCP_Domain	ZHCP domain name
ZHCP_OSAdev	ZHCP (internal network) real OSA device address

Note: XCAT_MN_vswitch is the z/VM Virtual Switch for the management network. The default value is XCATVSW2. Do *not* change it.

6.5.2 DMSSICMO COPY file for the controller node

CMA configures its services based on the properties in the DMSSICMO configuration file that contains the CMA and OpenStack parameters. This includes whether this CMA is a controller node (user functions and a compute node) or a compute node only (managed by another controller node). A script reads the data in this file and updates various OpenStack configuration files when the server starts.

Note: When modifying this file, note that it is case-sensitive. When editing it, always run the **case mixed i** command.

Example 6-24 shows a DMSSICMO COPY example of the use of CMA.

Example 6-24 DMSSICMO COPY example of using CMA

```

/*****
/* CMO User Configurable Settings                               */
/*****
cmo_admin_password      = "cmopass"
cmo_data_disk           = "LXBK01 LXBK02"
openstack_system_role   = "controller"
openstack_controller_address = "ip address"
openstack_zvm_diskpool  = "ECKD:yourname"
openstack_instance_name_template = "df1t1nx"
openstack_zvm_fcp_list  = "NONE"
openstack_zvm_timeout   = "300"
openstack_zvm_scsi_pool = "NONE"
openstack_zvm_zhcp_fcp_list = "NONE"
openstack_san_ip        = "NONE"
openstack_san_private_key = "NONE"
openstack_storwize_svc_volpool_name = "NONE"
openstack_storwize_svc_vol_iogrp  = "NONE"
openstack_zvm_image_default_password = "password"
openstack_xcat_mgt_ip             = "10.10.10.10"
openstack_xcat_mgt_mask           = "255.255.255.0"
openstack_zvm_xcat_master         = "your1parname"
openstack_zvm_vmrelocate_force    = "NONE"
openstack_default_network         = "9.76.61.1-9.16.8.29/2"
openstack_zvm_xcat_service_addr   = "9.16.8.23"
openstack_volume_enable_multipath = "FALSE"

```

The following configuration options are available:

cmo_admin_password	OpenStack controller default password, which is changed inside the guest operating system during initial configuration.
cmo_data_disk	VOLIDs for OpenStack controller data disk.
openstack_system_role	OpenStack controller role.
openstack_zvm_diskpool	IBM ECKD™ or FBA:DIRMAINT definition for OpenStack disk pool.
openstack_instance_name_template	z/VM user ID instance name template.
openstack_zvm_image_defaults_password	Default root password for deployed instances, which should be changed inside the guest operating system during initial configuration. Consider making this option AUTOONLY or LBYONLY.
openstack_zvm_xcat_master	xCAT Management Node name.

Note: For the SCSI environment, the value that is used for configuration option was `openstack_zvm_diskpool = "FBA:xxxxx"`.

6.6 CMA compute node

The compute node configuration allows the CMA to integrate into an OpenStack environment as a compute node. In this case, the z/VM drivers are not needed in the other OpenStack environment because that function is part of the OpenStack services on the compute node.

6.6.1 DMSSICNF COPY file for the compute node

The DMSSICNF COPY file contains several global attributes that can be modified to better fit your installation. In this case, you define it on the ITSOVM2 LPAR.

Note: When modifying this file, note that it is case-sensitive. When editing it, always run the **case mixed i** command.

Example 6-25 shows a DMSSICNF COPY example of the use of CMA.

Example 6-25 DMSSICNF COPY example of using CMA

```
/*
/* *****
/* XCAT server defaults
/* *****
XCAT_User      = "OPNCLOUD"           /* xCAT z/VM user ID
XCAT_Addr      = "10.10.10.30"        /* XCAT IP Address
XCAT_Host      = "xcat2"              /* xCAT hostname
XCAT_Domain    = ".cpolab.ibm.com"    /* Main name
XCAT_vswitch   = "XCATVSW1"          /* xCAT Vswitch name
XCAT_OSAdev    = "2126"              /* OSA address for xCAT
XCAT_zvmssid   = "rdbkzvm2"          /* xCAT z/VM system id
XCAT_notify    = "OPERATOR"          /* Notify when xCAT started
XCAT_gateway   = "10.10.10.1"        /* Network gateway IP addr.
XCAT_netmask   = "255.255.255.0"     /* Default network mask
XCAT_vlan      = "NONE"
XCAT_iso       = ""
XCAT_MN_Addr   = "ip address" /* xCAT mgmt node IP address */
XCAT_MN_vswitch = "XCATVSW2"         /* xCAT MN Vswitch name
XCAT_MN_OSAdev  = "2106"             /* OSA address for xCAT MN
XCAT_MN_gateway = "x.xx.x.1"         /* Network gateway IP addr.
XCAT_MN_Mask    = "255.255.240.0"    /* Netmask for xCAT MN
XCAT_MN_vlan    = "NONE"
XCAT_MN_admin   = "mnadmin"          /* MN administrator userid
XCAT_MN_pw      = "yourppsw"         /* MN admin password
/* (if NOLOG, userid cannot
/* ssh into XCAT MN)

/*
/* *****
/* ZHCP server defaults
/* *****
ZHCP_User      = "ZHCP"              /* zhcp z/VM user ID
ZHCP_Addr      = "10.10.10.40"        /* zhcp IP ADDRESS
ZHCP_Host      = "zhcp2"             /* zhcp hostname
ZHCP_Domain    = ".itso.ibm.com"     /* zhcp domain name
ZHCP_gateway   = "10.10.10.1"        /* Network gateway IP addr.
ZHCP_netmask   = "255.255.255.0"     /* Default network mask
ZHCP_vswitch   = "XCATVSW1"          /* zhcp Vswitch name
```

```
ZHCP_OSAdev = "2126" /* OSA address for zhcp */
ZHCP_vlan = "NONE"
/*****/
```

The following configuration options are available:

XCAT_Domain	OpenStack compute domain name
XCAT_OSAdev	XCATVSW1 (for xCAT internal network) real OSA device address
XCAT_zvmsysid	z/VM SYSID where OpenStack controller run (lowercase)
XCAT_MN_Addr	OpenStack controller (xCAT Management Node) IP address
XCAT_MN_OSAdev	XCATVSW2 (for management network) real OSA device address
XCAT_MN_gateway	OpenStack controller default gateway
XCAT_MN_netmask	OpenStack controller default netmask
XCAT_MN_pw	xCAT Management Node (mnadmin) default password
ZHCP_Host	ZHCP host name
ZHCP_Domain	ZHCP domain name
ZHCP_OSAdev	ZHCP (internal network) real OSA device address

Note: XCAT_MN_vswitch is the z/VM Virtual Switch for the management network. The default value is XCATVSW2. Do *not* change it.

6.6.2 DMSSICMO COPY file for the compute node

CMA configures its services based on the properties in the DMSSICMO configuration file, which contains the CMA and OpenStack parameters. This configuration includes whether this CMA is a controller node (user functions and a compute node) or a compute node only (managed by another ICM controller node). A script reads the data in this file and updates various OpenStack configuration files when the server starts.

Note: When modifying this file, note that it is case-sensitive. When editing it, always run the **case mixed i** command.

Example 6-26 shows a DMSSICMO COPY example of the use of CMA.

Example 6-26 DMSSICMO COPY example of using CMA

```
/*****/
/* CMO User Configurable Settings */
/*****/
cmo_admin_password = "yourppw"
cmo_data_disk = "valid1 valid2 valid3 valid4"
openstack_system_role = "compute"
openstack_controller_address = "ip address"
openstack_zvm_diskpool = "ECKD:xxxx"
openstack_instance_name_template = "osp%05x"
openstack_zvm_fcp_list = "NONE"
openstack_zvm_timeout = "300"
openstack_zvm_scsi_pool = "NONE"
openstack_zvm_zhpc_fcp_list = "NONE"
openstack_san_ip = "NONE"
```

```

openstack_san_private_key          = "NONE"
openstack_storwize_svc_volpool_name = "NONE"
openstack_storwize_svc_vol_iogrp   = "NONE"
openstack_zvm_image_default_password = "yourppw"
openstack_xcat_mgt_ip              = "NONE"
openstack_xcat_mgt_mask             = "NONE"
openstack_zvm_xcat_master           = "xcat1"
openstack_zvm_vmrelocate_force      = "NONE"
openstack_default_network           = "NONE"
openstack_zvm_xcat_service_addr     = ""
openstack_volume_enable_multipath   = "FALSE"

```

The following configuration options are available:

cmo_admin_password	OpenStack controller default password
cmo_data_disk	VOLIDs for OpenStack controller data disk
openstack_system_role	OpenStack controller role
openstack_zvm_diskpool	ECKD or FBA:DIRMAINT region for OpenStack disk pool
openstack_instance_name_template	z/VM user ID instance name template
openstack_zvm_image_defaults_password	Default root password for deployed instances
openstack_zvm_xcat_master	xCAT Management Node name

Note: For the SCSI environment, the value that is used for the configuration option was `openstack_zvm_diskpool = "FBA:xxxxx"`.

To add compute nodes for other LPARs on your environment, repeat the definitions that are described in this chapter by updating the specific information for the LPAR.

For more information about CMA for OpenStack Newton, see *Enabling z/VM for OpenStack (Support for OpenStack Newton Release)*, SC24-6251, and *Systems Management Application Programming (6.4)*, SC24-6327.

6.7 CMA installation

The instructions that are presented in this section are for installing the z/VM 6.4 CMA on an existing z/VM 7.1 system. The CMA running on 7.1 requires SMAPI running at the 6.4 version. You cannot run the 7.1 version of SMAPI at the same time as the CMA. The 7.1 version of SMAPI remains installed and available if you want to run the 7.1 version of SMAPI in the future.

The z/VM 6.4 CMA running on z/VM 7.1 requires that the z/VM 6.4 version of SMAPI is operational and the CMA-specific configuration values and files be available on z/VM 7.1. This process includes the following tasks:

- ▶ Install (restore) SMAPI 6.4 on your 7.1 system (leaving your 7.1 SMAPI code intact) and configure your SMAPI servers to use the 6.4 level of SMAPI.
- ▶ Create the CMA user ID (OPNCLOUD) and allocate the necessary resources. Install (restore) the CMA code.
- ▶ Authorize OPNCLOUD with your ESM, SMAPI, and Directory Manager.

- Configure SMAPI and OPNCLOUD as documented and appropriate for your specific needs.

To install and run CMA Newton, you need the MDiskS that are listed in Table 6-1.

Table 6-1 Required MDiskS

Owner	Virtual address	Volume use	ECKD cylinders
OPNCLOUD	191	OPNCLOUD 191 disk	1
OPNCLOUD	101	CMA Newton first System disk	3338
OPNCLOUD	102	CMA Newton second System disk	3338
MAINT710	102	Packed copy of CMA101 ECKPDPACK file	3338
MAINT710	103	Packed copy of CMA102 ECKPDPACK file	3338
MAINT710	104	Unpacked copy of CMA101 ECKD file	3338
MAINT710	105	Unpacked copy of CMA102 ECKD file	3338
MAINT710	106	Packed copy of MNT1193 ECKDDPACK file	500
MAINT710	107	Unpacked copy of MNT1193 ECKD file	500
MAINT	1193	Restored MAINT 1193 disk	500

If you use a CP directory management tool, such as DirMaint, use the appropriate directory management tool commands to add this MDisk. If you are not using a directory management tool, use XEDIT to change your CP user directory by using your local change management procedures.

Note: To get the CMA Newton package, contact your IBM representative or IBM account team because this option is restricted to ICP customers.

The following materials are available from your IBM account team:

- MNT1193 ECKDPACK file. This file is a DDR image to be restored on the new MAINT 1193 virtual disk you create.
- CMA101 ECKDPACK file. This file is a DDR image to be restored on the new OPNCLOUD 0101 virtual disk you create.
- CMA102 ECKDPACK file. This file is a DDR image to be restored on the new OPNCLOUD 0101 virtual disk you create.

Note: The files must be transferred to your z/VM system as binary, fixed-length record format files with a logical record length of 1024.

6.7.1 Initial set-up

Complete the following steps to create the OPNCLOUD user ID and disks that are required for the installation:

1. Use the definitions that are found Appendix H, “OPNCLOUD Directory Entry” of *Systems Management Application Programming Version 6 Release 4*, [SC24-6234](#).
2. Using Table 6-1 on page 185, create the required MDiskS for MAINT710 and MAINT. The OPNCLOUD 101, 102, and 191 MDiskS were created as part of adding the OPNCLOUD user ID to your system. If these MDiskS were not created, create them now.

6.7.2 Installing SMAPI 6.4 on your 7.1 system

This step installs a copy of z/VM 6.4 SMAPI on your z/VM 7.1 system. The 7.1 SMAPI code is not overwritten or removed from your system. The 6.4 version of SMAPI is on the MAINT 1193 disk. The 7.1 SMAPI code remains on the MAINT 193 disk.

Complete the following steps to restore the contents of the MAINT 1193 disk:

1. Run the **CP LINK * 106 106 MR** command from MAINT710.
2. Run the **CP LINK * 107 170 MR** command from MAINT710.
3. Run the **CP LINK MAINT 1193 1193 MR** command from MAINT710.
4. Run the **FORMAT 106 T** command from MAINT710:
 - Answer 1 when prompted to format.
 - Answer MNT106 when prompted to provide a label.
5. Run the **FORMAT 107 U** command from MAINT710:
 - Answer 1 when prompted to format.
 - Answer MNT107 when prompted to provide a label.
6. Run the **FORMAT 1193 V** command from MAINT710:
 - Answer 1 when prompted to format.
 - Answer MN1193 when prompted to provide a label.
7. Run the **CP DETACH 106** command from MAINT710.
8. Use FTP to move the MNT1193 ECKDPACK file from your workstation to z/VM.

Note: All ECKDPACK files must be transferred as a binary, fixed-length record format with a logical record length of 1024 file.

9. From your workstation command prompt, enter the following information:
 - FTP ipaddr_VM_system
 - USER MAINT710
 - Password
10. Run the **CP DETACH 106** command from MAINT710.
11. Use FTP to move the MNT1193.ECKDPACK file from your workstation to z/VM.
12. Use FTP to move the MNT1193 ECKDPACK file from your workstation to z/VM.
13. From your workstation command prompt, enter the following information:
 - FTP ipaddr_VM_system
 - USER MAINT710
 - Password

The DDRREST command displays out that is similar to the output that is shown in Example 6-27.

Example 6-27 DDRREST messages

```
z/VM DASD DUMP/RESTORE PROGRAM
HCPDDR698I DATA DUMPED FROM OCSYS1 TO BE RESTORED
HCPDDR697I NO VOL1 LABEL FOUND
RESTORING MT1193
DATA DUMPED mm/dd/yy AT hh.mm.ss GMT FROM MNT1193 RESTORED
INPUT CYLINDER EXTENTS OUTPUT CYLINDER EXTENTS
START STOP START STOP
0 499 0 499
END OF RESTORE
BYTES RESTORED xxxxxxxxx

END OF JOB
```

6.7.3 Installing the CMA files on your z/VM 7.1 system

Log on to the MAINT710 user ID. You use the MDiskS that were defined to hold the restorable CMA images (virtual device addresses: 102-105). These four MDiskS are owned by the MAINT710 user ID and are defined as the sizes listed in the Table 6-1 on page 185. They contain copies of the CMA images that are provided by your IBM representatives.

These copies must exist only on your z/VM 7.1.0 system until the CMA is restored and running. If you are in a multi-member SSI, these MDiskS can be shared across the MAINT710 members in the SSI cluster.

Note: To get the CMA Newton package, contact your IBM representative or IBM account team because this option is restricted to ICP customers.

After defining these MDiskS to the MAINT710 user ID, make them accessible to MAINT710. Complete the following steps:

1. Run the **CP LINK * 102 102 MR** command.
2. Run the **CP LINK * 103 103 MR** command.
3. Run the **CP LINK * 104 104 MR** command,
4. Run the **CP LINK * 105 105 MR** command.
5. Run the **FORMAT 102 T** command:
 - When prompted, respond 1.
 - When prompted again, respond MNT102.
6. Run the **FORMAT 103 U** command:
 - When prompted, respond 1.
 - When prompted again, respond MNT103.
7. Run the **FORMAT 104 V** command:
 - When prompted, respond 1.
 - When prompted again, respond MNT104.
8. Run the **FORMAT 105 W** command:
 - When prompted, respond 1.

- When prompted again, respond MNT105.
9. Run the following commands:
 - CP DETACH 102
 - CP DETACH 103
 10. Use FTP to move the CMA files from your workstation to z/VM. These files must be transferred as binary, fixed-length record format with a logical record length of 1024 files.

From your workstation command prompt, enter the following commands, one at a time:

1. FTP ipaddr_VM_system
2. USER MAINT710
3. password
4. BIN
5. QUOTE SITE FIXRECFM 1024
6. CD MAINT710.102
7. PUT CMA101.ECKDPACK CMA101.ECKDPACK
8. CD MAINT710.103
9. PUT CMA102.ECKDPACK CMA102.ECKDPACK
10. QUIT

Back on the z/VM MAINT710 user ID, copy the packed CMA image files to unpacked image files using the following commands:

1. CP LINK * 102 102 RR
2. ACCESS 102 T
3. CP LINK * 103 103 RR
4. ACCESS 103 U
5. COPYFILE CMA101 ECKDPACK T CMA101 ECKD V (UNPACK OLDDATE
6. COPYFILE CMA102 ECKDPACK U CMA102 ECKD W (UNPACK OLDDATE

6.7.4 Restoring the CMA files

Now that the CMA image files are on your z/VM 7.1.0 system, you must restore the image files to the OPNCLOUD 101 and 102 minidisks. While still logged on to MAINT710, enter the following commands:

1. DET 101-102
2. LINK OPNCLOUD 101 101 MR
3. LINK OPNCLOUD 102 102 MR
4. ACCESS 193 R
5. ACCESS 104 V
6. ACCESS 105 W
7. DDRREST 101 CMA101 ECKD V
8. DDRREST 102 CMA102 ECKD W

The use of the **DDRREST** command displays output that is similar to the output that is shown in Example 6-28.

Example 6-28 DDREREST messages

```
z/VM DASD DUMP/RESTORE PROGRAM
HCPDDR698I DATA DUMPED FROM OCSYS1 TO BE RESTORED
HCPDDR697I NO VOL1 LABEL FOUND
RESTORING OCSYS1
DATA DUMPED mm/dd/yy AT hh.mm.ss GMT FROM OCSYS1 RESTORED
INPUT CYLINDER EXTENTS OUTPUT CYLINDER EXTENTS
```

```
START STOP START STOP
0 3337 0 3337
END OF RESTORE
BYTES RESTORED 2467590380
END OF JOBExample on page 187
```

Note: Be careful to verify the extents that are copied and bytes restored values.

After restoring the CMA to the OPNCLOUD 101 minidisk and 102 minidisk, run the **DETACH 101-102** command to relinquish control of the MDisk.

6.7.5 Configuring to use CMA 6.4 (Newton)

The following configuration changes also must be made to your z/VM 7.1 system for the CMA:

- Make a copy of the 6.4 version of SSPSTART EXEC and OPNCLOUD SAMPPROF by running the following commands from MAINT710:

```
LINK OPNCLOUD 191 91 MR
LINK MAINT 1193 1193 RR
ACCESS 91 G
ACCESS 1193 H
COPYFILE SSPSTART EXEC H SSPSTART EXEC G (OLDDATE
COPYFILE OPNCLOUD SAMPPROF H PROFILE EXEC G (OLDDATE
RELEASE G (DETACH
RELEASE H
```

- Point the new MAINT 1193 MDisk.

The MAINT 193 disk is at the z/VM 7.1 level. However, the CMA requires the files on the 193 disk be at the z/VM 6.4 level. To do this, the following USERS must be updated to have their virtual 193 disk use the MAINT 1193 disk:

- VSMGUARD
- VSMWORK1
- VSMWORK2
- VSMWORK3
- VSMWORKx (if you added SMAPI worker servers)
- VSMREQUIU
- VSMREQIN
- VSMREQI6
- LOHCOST
- OPNCLOUD

These user IDs are all IDENTITY users in an SSI-enabled directory. The appropriate SUBCONFIG entry should be updated for each IDENTITY.

Change from:

```
LINK MAINT 0193 0193 RR
```

To:

```
LINK MAINT 1193 0193 RR
```

- The startup of OPNCLOUD might generate several error messages that include the text “failed to get suitable CP”. These messages are informational only and can be safely ignored.

For more information about how to configure SMAPI and CMA, see *Systems Management Application Programming version 6 release 4, Enabling z/VM for OpenStack (Support for OpenStack Newton Release) version 6 release 4, SC24-6253*.

Stop using CMA 6.4 (Newton)

To stop using the 6.4 CMA on z/VM 7.1 and switch to the 7.1 version of SMAPI, the following servers should have their directory entries changed as described next:

- ▶ VSMGUARD
- ▶ VSMWORK1
- ▶ VSMWORK2
- ▶ VSMWORK3
- ▶ VSMWORKx (if you added SMAPI worker servers)
- ▶ VSMREQUIU
- ▶ VSMREQIN
- ▶ VSMREQI6
- ▶ LOHCOST
- ▶ OPNCLOUD

These user IDs are all IDENTITY users. In an SSI-enabled directory, the appropriate SUBCONFIG entry should be updated for each IDENTITY.

Change from:

```
LINK MAINT 1193 0193 RR
```

To:

```
LINK MAINT 0193 0193 RR
```

No other changes are required. Restart SMAPI by FORCEing and XAUTOLOGging VSMGUARD.

6.8 Securing your cloud components

A Cloud on z/VM environment might use several components. It is important to protect each of the components.

The cloud components on z/VM can be protected with most of what is described in Chapter 3, “IBM z/VM hypervisor” on page 25, but that does not remove the need for an ESM. When using RACF to control your cloud resources, grant the VMs only the bare minimum privileges required to perform their intended tasks.

As described in 5.1.1, “Least privilege principle” on page 108, do not allow the VMs to exceed the scope of their responsibility.

It is important to have your company’s security policy job roles relate to the cloud, such as a cloud administrator and a cloud auditor. Make sure the job roles that are related to the cloud also have their accesses described in the security policy, and that those accesses are implemented across the cloud environment.

Integrating the identity management across the cloud environment makes it easy to manage. Since z/VM 6.3, OpenStack Keystone is supported for installation-wide authentication and authorization to OpenStack Services.

The use of the identity integration brings some important capabilities to the cloud environment and z/VM, including authenticating user and password requests against multiple back ends, such as SQL or LDAP, as described in 8.2, “Lightweight Directory Access Protocol” on page 246.

The following key service capabilities also are available:

- ▶ Token: Validates and manages tokens for user authentication
- ▶ Catalog: Allows for endpoint registry of available services.
- ▶ Policy: Authorizes API requests, and others, such as domain, project, and user models with role-based access control (RBAC) for access compute, storage, and networking.

Table 6-2 lists the security mechanisms that are available when a cloud environment is deployed on top of z/VM. Because the cloud can be composed of several components, careful attention should be given to each component to prevent eventual security breaches across your environment.

Table 6-2 Security mechanisms in a private cloud on z/VM environment

z/VM and CMA cloud layers	Security mechanism	Risks addressed
CMA (OpenStack Controller)	<ul style="list-style-type: none"> ▶ Projects ▶ Roles and RBAC ▶ Identity management 	<ul style="list-style-type: none"> ▶ Account hijacking ▶ Malicious insiders
OpenStack (Compute Node)	<ul style="list-style-type: none"> ▶ Tenancy ▶ HTTPS (encryption) 	<ul style="list-style-type: none"> ▶ Insecure APIs ▶ Denial of service
xCAT	<ul style="list-style-type: none"> ▶ Identity management ▶ SSH (encryption) 	<ul style="list-style-type: none"> ▶ Insecure APIs ▶ Abuse and nefarious use
SMAPI	<ul style="list-style-type: none"> ▶ RBAC by API ▶ TLS (encryption) 	<ul style="list-style-type: none"> ▶ Insecure APIs ▶ Malicious insiders
DirMaint for z/VM	<ul style="list-style-type: none"> ▶ Resource access control ▶ Auditing 	<ul style="list-style-type: none"> ▶ Data loss ▶ Insufficient due diligence
z/VM (CP with RACFVM)	<ul style="list-style-type: none"> ▶ Guest isolation ▶ Privilege classes ▶ RBAC ▶ Security zones ▶ Auditing (SMF) 	<ul style="list-style-type: none"> ▶ Data breaches ▶ Account hijacking ▶ Abuse and nefarious use ▶ Insufficient due diligence ▶ Shared technology issues

6.8.1 Security considerations inherent in a cloud environment

In this section, we describe guidelines for security and privacy in a cloud computing environment.

Governance

Extend organizational practices that pertain to the policies, procedures, and standards that are used for application development and service provisioning in the cloud, and the design, implementation, testing, use, and monitoring of deployed or engaged services.

Put in place audit mechanisms and tools to ensure that organizational practices are followed throughout the system lifecycle.

Compliance

Understand the various types of laws and regulations that impose security and privacy obligations on the organization and potentially affect cloud computing initiatives, particularly those involving data location, privacy and security controls, records management, and electronic discovery requirements.

Review and assess the cloud provider's offerings regarding the organizational requirements to be met and ensure that the contract terms adequately meet the requirements. Ensure that the cloud provider's electronic discovery capabilities and processes do not compromise the privacy or security of data and applications.

Trust

Ensure that service arrangements have sufficient means to allow visibility into the security and privacy controls and processes that are used by cloud computing and their performance over time. Establish clear, exclusive ownership rights over data.

Institute a risk management program that is flexible enough to adapt to the constantly evolving and shifting risk landscape for the lifecycle of the system. Continuously monitor the security state of the information system to support on-going risk management decisions.

Architecture

Understand the underlying technologies that the cloud computing uses to provision services, including the implications that the technical controls involved have on the security and privacy of the system, over the full system lifecycle, and across all system components.

IdEA Mgmt

Ensure that adequate safeguards are in place to secure authentication, authorization, and other identity and access management functions, and are suitable for the organization.

Software isolation

Understand virtualization and other logical isolation techniques that the cloud provider employs in its multi-tenant software architecture, and assess the risks involved for the organization.

Data protection

Evaluate the suitability of the cloud computing's data management solutions for the organizational data concerned and the ability to control access to data to secure data while at rest, in transit, and in use, and to sanitize data.

Consider the risk of collating organizational data with that of other organizations whose threat profiles are high or whose data collectively represent significant concentrated value. Fully understand and weight the risks that are involved in cryptographic key management with the facilities that are available in the cloud environment and the processes that are established by the cloud provider.

Availability

Understand the contract provisions and procedures for availability, data backup and recovery, and disaster recovery, and ensure that they meet the organization's continuity and contingency planning requirements. Ensure that during an intermediate or prolonged disruption or a serious disaster, critical operations can be immediately resumed, and that all operations can be eventually reinstituted in a timely and organized manner.

Incident response

Understand the contract provisions and procedures for incident response and ensure that they meet the requirements of the organization. Ensure that the cloud computing has a transparent response process in place and sufficient mechanisms to share information during and after an incident.

Also, ensure that the organization can respond to incidents in a coordinated fashion with the cloud environment in accordance with their respective roles and responsibilities for the computing environment.

6.8.2 Security tips for the cloud

In this section, we describe some security tips for the cloud.

Do not forget the basics

Consider the following points:

- ▶ Access controls and incidence response do not go away in the cloud.
- ▶ Be mindful of asset-tracking, data flow, and change management.
- ▶ Your data is of varying classifications; therefore, use multi-tenant features when pertinent.

Use cloud to enhance security

Consider the following points:

- ▶ Cloud offers rapid scale of environments and networking.
- ▶ Employ Security as a Service for varying definitions of security, such as threat and intrusion detection, and vulnerability scanning mechanisms.
- ▶ Standardizing means less variability, which can mean fewer surprises.

Understand your cloud's capabilities

Consider the following points:

- ▶ Determine whether your provider is part of your company or a third party.
- ▶ Establish clear roles and communication paths for escalations and incident management.
- ▶ Understand where the data must be stored (for regulatory and geopolitical reasons) and who is the owner and can access it.



Securing IBM Cloud Private and Microservices on LinuxONE

Today's work style and the rapid changes in customer demands require faster time for deployment and productivity. Enterprises must be flexible while catching up to the continuous changes of such demands and the rapid technology change while ensuring that they keep their systems secure against external and internal threats to ensure their reputation is kept intact and comply with any regulations imposed. Most organizations are targeting this change: To have an agile cloud infrastructure while ensuring it is both reliable, highly available and, most importantly, secure.

This chapter provides an overview of DevOps, microservices and their deployment on top of a Kubernetes managed infrastructure. This chapter also describes how to securely deploy different microservices. We describe an example of the security we employed in our lab environment when using IBM Cloud Private running on LinuxONE

Additionally, we discuss Kubernetes specifics and how it orchestrates containers and controls the data flow between them. Finally, an introduction to IBM Cloud Private and implementation guidelines on LinuxONE is provided so that customers can leverage their existing Cloud portfolio under an industry leading security platform.

This chapter includes the following topics:

- ▶ 7.1, "Security in DevOps" on page 196
- ▶ 7.2, "Introduction to microservices" on page 196
- ▶ 7.3, "Managing containers by using Kubernetes" on page 202
- ▶ 7.4, "Containers management at scale" on page 213
- ▶ 7.5, "IBM Cloud Private overview" on page 216
- ▶ 7.6, "IBM Cloud Private on LinuxONE" on page 223
- ▶ 7.7, "IBM Cloud Automation Manager" on page 239

7.1 Security in DevOps

DevOps is a culture that brings development and operations teams together, including agile and lean methodologies to the software development lifecycle. More and more, cloud development projects use DevOps as their preferred method of development because of its speed to delivery and lower costs for development, testing, deployment, and operations.

The requirement to produce higher-quality software and achieve lower costs in a short period is critical for organizations during their shift to the Cloud. However, as with any organizational change, a security mindset must be present always. It is important that DevOps implements security practices that are aligned with the corporate security policy in place, in such a way that every step during the management of your Cloud infrastructure is done securely and complies with both internal and external regulations.

DevOps security refers to the practice of securing the entire DevOps environment through strategies, processes, and technology that is used. Security should be included in each phase of the DevOps lifecycle, including design, build, test, release, support, and maintenance. Often referred to as *DevSecOps*, it aims to improve security through collaboration and shared responsibility that covers the entire DevOps workflow.

When creating a secure cloud offering, the following facets should be integrated into the DevOps processes within an organization:

- ▶ Secure engineering ensures that the products and services are developed and built with strong security and privacy controls and operate in compliance with accepted international, national, governmental, industry, and regional security standards.
- ▶ Secure deployment and operations ensure that the cloud, runtimes, and applications are deployed securely, checked regularly for security configuration and hygiene, tested for security vulnerabilities, and are frequently patched and updated against known security vulnerabilities, and remain under a vendor supported level.
- ▶ Separation of duties ensures that users have only the minimum access as required to perform their job roles, as defined in the principle of least privilege.
- ▶ Availability and business continuity management ensures that the infrastructure, runtime components, and management components are highly available.
- ▶ Security evaluation and learning ensures that the security functions and properties in the delivered code and services are addressed as threats evolve and new vulnerabilities arise.

To fully understand the security components that are needed for secure cloud development, deployment, and operations, see this [web page](#).

7.2 Introduction to microservices

The microservices architecture is rapidly gaining adoption and popularity across organizations because of its improved productivity capabilities and speed to market. It directly contrasts with the monolithic architecture, on which an application is built as a single entity and requires an update to the entire application code whenever developers must alter the system.

This section provides an introduction to the microservices architecture and service discovery. We also discuss commonly implemented security practices when deploying a microservice oriented application.

7.2.1 Microservice architecture

The microservice architecture structures the application implementation as a set of multiple components. These components are services, and the relationship between them is done through communication protocols (typically HTTP APIs) that enable those services to collaborate. The collection of such services structure an entire application as a collection of loosely coupled¹, independently deployable services.

By breaking the application into smaller, self-capable components, each of these components focuses on completing one task that, in turn, represents a small business requirement or feature.

Several benefits are realized by adopting a microservices oriented architecture. The development and maintenance of an application often is handled by several developers that might be geographically dispersed or distributed across several functional teams. Microservices enable teams to work in a more autonomous way, which allows for continuous updates, improved maintenance, and higher productivity.

Microservices components also have higher scalability than traditional applications. Because the communication among components of an application occurs by using APIs, it is possible to scale up and down such components to achieve higher availability and resilience rate for applications. In that sense, a failed component does not cause the entire application to be unavailable. Finally, because microservices are small lightweight components that together form a working application, the overall underlying system performance is improved.

Figure 7-1 shows a sample application that uses a microservices architecture. Every component of the application is broken down into smaller, self-capable parts that interact with each other according to business needs.

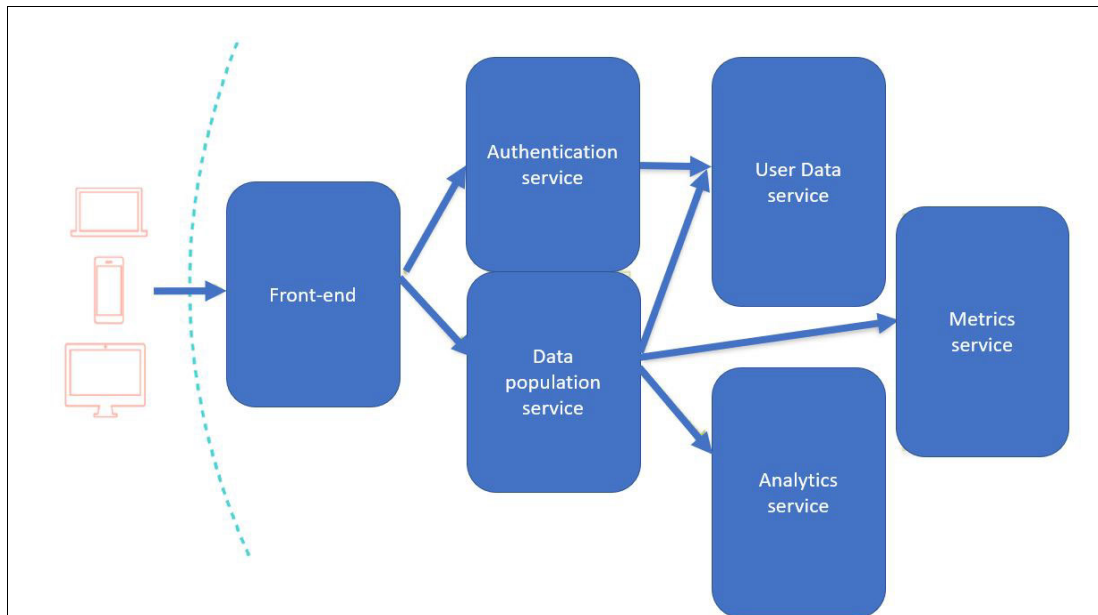


Figure 7-1 Sample application using microservices architecture

¹ A loosely coupled system allows for its components to be replaced without making updates to other components. That is, it reduces the inter-dependencies between other components and achieves higher flexibility and maintainability.

One of the common implementation problems that is faced by organizations during the deployment of a microservice-oriented application relates to the increased complexity that is caused when cross dependency is introduced between its inter-connected components. In that sense, it is important to understand the relationship among such components to properly design and implement truly self-capable services.

For example, a common problem that is faced during microservice implementations occurs when two or more components share access to a single database, which leads to duplicated data sharing and increased database loads. Similarly, two separate components that must communicate with each other also might be the root cause for increased latency rates within an application.

Microservices are small in essence. However, a fundamental understanding of the relationship between each service is essential to avoid problems during the development process. Always consider that a microservice must deliver sufficient capabilities to its related components. When designing microservices with this mindset, it is possible to eliminate several dependency problems that commonly arise.

Almost every, if not all, microservice-oriented applications run on top of containers. Because each component is relatively small when compared to traditional monolithic applications, it allows for easier infrastructure management and higher scalability rates.

Spinning-up several microservices also has the benefit of achieving application high-availability. Therefore, the development of microservices is considered strategic for organizations during their shift to the cloud. In that sense, the IBM LinuxONE Emperor II can scale up to 2 million containers within a single hardware footprint in such a way that the platform is designed to meet the current and future demands for microservices applications.

Because most microservices interactions occur by using HTTP APIs, the IBM LinuxONE Emperor II can also serve up to 30 billion web data requests a day without any performance impact, while moving data faster than alternative platforms with 2.1x higher data processing throughput. All of these factors make the IBM LinuxONE family the ideal choice for organizations that want to pursue cloud-oriented services.

Figure 7-2 on page 199 shows the scaled version of our microservices application. This ideal scenario is best for organizations to pursue during their shift to a cloud-oriented infrastructure.

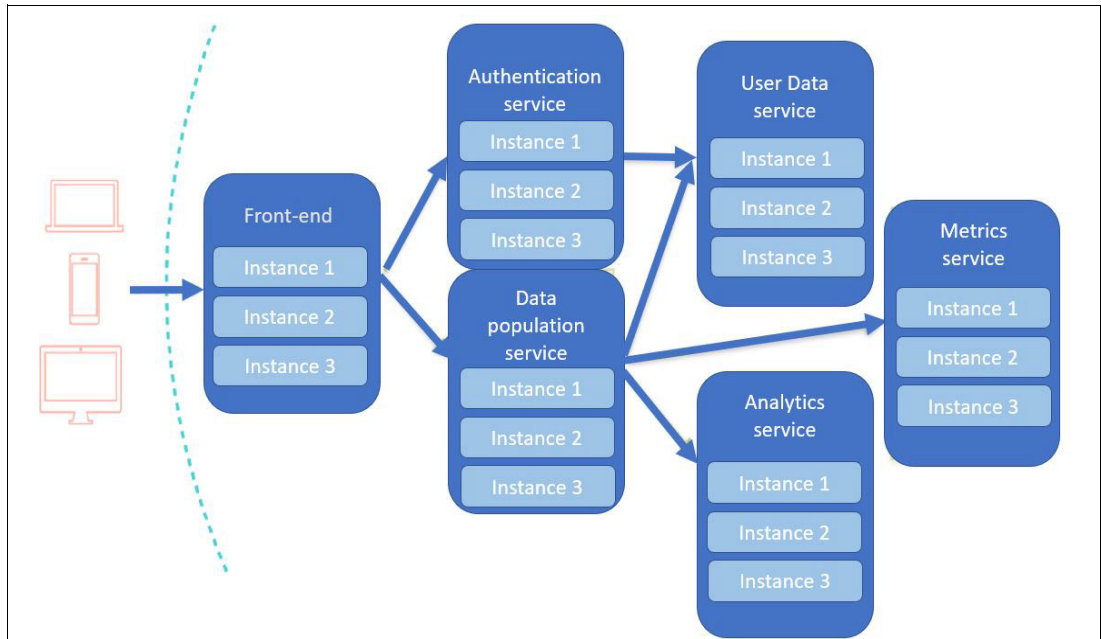


Figure 7-2 Scaled version of microservices application

7.2.2 Service discovery

In the traditional distributed computing model, the components of an application, such as a web server, middleware, and databases, communicate with each other by using static network routes. That is, the components are aware of the exact location to procure when an external resource is required.

However, because microservices can coexist across different containers, servers, and even different locations, a mechanism to find the dynamically assigned components of an application and route traffic to and from them must be implemented. Moreover, when scaling microservices, it is important to ensure that the traffic is balanced across all replicas within the cluster.

Service discovery refers to a process in which each microservice can reach its peers without being aware about their location. This modern cloud concept is critical for the successful deployment of microservices. It uses a *service registry* to store, update, and provide (whenever queried) the routing to available services.

Because of the nature of microservices instances to be scaled up and down according to the business demands, it is critical that the service registry is highly available and constantly updated to prevent it from affecting a business. Similarly, service registries typically use health check mechanisms to ensure that the registered services are still active and ready for servicing.

Finally, a load balancer that is responsible for querying the service registry is used. It is then responsible for balancing the load across the replicas in such a way that the load is spread across the entire cluster. Because of its critical role, the load balancer also should be highly available in such a way that the entire network flow can occur without interruption to the business.

Service discovery in Kubernetes

When Kubernetes is used to orchestrate containers, the deployment of microservices is greatly simplified. By default, whenever a service is created on a Kubernetes cluster, it automatically assigns an IP address to it.

This process occurs because a process that is called *kube-proxy* is present on each managed node within the cluster. When scaling up services across multiple nodes, the kube-proxy process is responsible for ensuring that the services are available to other components. It is also responsible for ensuring that requests are delivered to its respective microservices.

Another important component in Kubernetes is the etcd daemon. The etcd is a key-value store service that is used to store configuration data. The configuration data is then made available to each node within a cluster, which allows for service discovery to occur. Kubernetes also provides APIs that can be used by applications to retrieve several configuration details of its managed services.

Kubernetes eases the deployment of microservices because it automatically provides the role of a service registry. Kubernetes also provides the following mechanisms to allow service discovery to occur:

- ▶ By defining environment variables that exposes the IP address and port that are related to a service, Kubernetes automatically load balances all incoming traffic to their respective replicas whenever a request is received through its respective service address.
- ▶ The DNS cluster add-on often is the preferred way to achieve service discovery. When this cluster is used, services can be discovered by using native Linux tools, such as **nslookup** and **dig** commands, which allow for easier coding and integration.

Kubernetes automatically provides service registry mechanisms and abstracts a great part of the service discovery steps in such a way that the implementation of microservices can be made simpler without developers worrying about the networking and load balancing layers during application development.

7.2.3 Securing your microservices application

Many internal and external security considerations must be addressed when rolling out microservices applications throughout your infrastructure.

Internal considerations

Internal considerations refer to what you have within your own data center or network and include the following examples:

- ▶ JSON Web Tokens (JWT): JWTs are becoming the prevailing standard for representing claims between two parties. These open standard tokens are the foundation for the OAuth/OAuth2 protocols and the OIDC framework.

The OAuth/OAuth2 protocols are used to provide secure authorization, token expiration, and access token revocation.

OIDC provides a secure authentication layer on top of OAUTH.

The propagation of identity between microservices, and the resulting authorization decisions, are predominantly based on JWTs as opposed to proprietary approaches or other “heavier weight” token standards, such as SAML.

- ▶ Transport Layer Security (TLS) remains the predominant method of securing HTTP communications (traffic) between all components, including between microservices.

- ▶ **API keys and shared secrets:** APIs and microservices are often further secured by using non-PKI encryption that uses symmetric cryptography. Symmetric cryptography requires both parties to use the same key (that is, a “shared secret”) to encrypt and decrypt their messages.

If you are planning to use API keys to authorize and validate calls that a microservice receives, be sure to follow secure practices for establishing and distributing the keys, and to regularly update (rotate) the keys. Otherwise, your security can be compromised if a rogue employee or hacker somehow learns the keys.

- ▶ **Whitelisting:** Application or microservice whitelisting is the practice of specifying a list of approved applications and services that are permitted to use your microservices. Whitelisting is performed in addition to other practices, such as two-way TLS. Consider whitelisting for environments that require exceptionally high security.
- ▶ **Blacklisting:** It specifies specific applications, microservices, or servers that cannot access your microservice. Blacklisting is typically implemented by your network group, independent of specific applications or services as a means to restrict rogue sites or users from accessing your IT environment.
- ▶ **Toolchains:** Securing your DevOps toolchain is of critical importance in a microservices. DevOps is critical for microservices.
- ▶ **Command Line Interfaces (CLIs):** Similar to your DevOps toolchain, ensure that any CLI tools you use are secured according to the tool’s capabilities or by your workstation or server security controls.
- ▶ **Automate wherever possible.** Social engineering and human behavior weaknesses are the leading ways to compromise a system. Automation typically includes the use of security scanners that monitor containers for vulnerabilities and include an automated security update policy.

External considerations

External considerations refer to the communication with other users or systems that are on the internet and include the following examples:

- ▶ **TLS is used to encrypt traffic.**
- ▶ **Exposed publicly or only behind firewalls:** When exposing microservices as APIs, the architectural placement of your API gateway can be in front of or behind your firewalls. When in front, your gateway must be hardened. The use of reverse proxies are a common practice to permit access only to specific resources within a demilitarization zone. Moreover, consider exposing only the bare minimum set of resources that are required for users to use your application.

7.3 Managing containers by using Kubernetes

This section provides an introduction to containers, the key points behind containers, and the differences between containers and virtual machines. It also introduces Kubernetes for container orchestration.

7.3.1 Introduction to containers

Containers represent a major technology shift for infrastructure management and application development. It aims to simplify several application development activities while providing better system resources utilization. Containers are a critical component that are present on modern clouds in such a way that the union of containers with microservices applications is the key component for an organization's digital success.

Similarly, the IBM LinuxONE platform is ready to meet the current and future growing container requirements. When LinuxONE is used as the platform for hosting your cloud services, it achieves excellent and unmatched scalability numbers when compared to x86 alternatives. Because of these reasons, the use of a container-based infrastructure also helps to reduce costs by improving the overall infrastructure consolidation process, making infrastructure management easier, and speeding up applications to market.

Conceptually, a *container image* encapsulates that required runtime components of an application, such as libraries, frameworks, and configuration files, along with the application binaries and code. By doing so, it instantiates much smaller and self-capable containers that can be scaled up on a single host, or scaled out across multiple systems. Because of these factors, containers become the de facto infrastructure of choice for building and deploying microservice applications.

Images and containers

A container image is a collection of immutable layers that together contain all packaged components that are required to run an application. An image can be shared across multiple functional teams by using public or private registries. They also can be easily maintained and updated to different needs. An image primarily consists of the following components:

- ▶ An application
- ▶ Its dependencies, libraries, and other binary files
- ▶ Configuration files that are needed to run it

After an image is created and is ready to be distributed, it is then *instantiated* on what are called containers. That is, while an image is a static representation of an application, a container represents a running instance of such image. This method ensures application consistency across various environments in such a way that the roll-out time from development to production is greatly enhanced. Similarly, it also dramatically reduces human errors during the application configuration, deployment, and implementation steps.

The practice of creating and distributing container images and running containers from them is also referred as *containerization*.

7.3.2 Containers versus virtual machines

A common misconception that occurs when starting the study of container-based technologies is that containers are substitutes to the traditional virtualization model. It is important to understand the benefits of each technology to build a more effective cloud infrastructure.

Containers act on a different layer than the traditional virtualization model does. Both technologies also have much in common in such a way that one complements each other.

Containers and virtual machines have similar resource isolation and allocation benefits, but function differently. Virtualization achieves infrastructure consolidation, reduces data center space, and allocates hardware resources across multiple operating systems in a single hardware footprint. Containers better use operating system resources and achieve high scalability, portability, and efficiency levels.

Virtual machines run on top of a hypervisor and share the hardware resources that are provided by it. Containers run on top of an operating system (typically Linux) and use the resources that are available within that operating system. Containers also share the underlying kernel from the host operating system; virtualized operating systems can run separate kernel levels as needed.

However, in the traditional virtualization model, different applications run on separated and dedicated operating systems. Each operating system uses a specific amount of resources as defined by the hypervisor.

As the infrastructure grows, a classic scenario occurs: Several operating systems are being virtualized, each one with its own CPU and memory requirements, and each one loads the entire operating system stack, such as daemons, libraries, and the required programs into memory. This cascade effect not only causes the hypervisor resources to be wasted, but also to be underutilized. Containers address this scenario because they are typically much more lightweight than virtualizing an entire operating system. Therefore, several containers can be run within a single operating system footprint.

Figure 7-3 shows a comparison between the structure of virtual machines and containers.

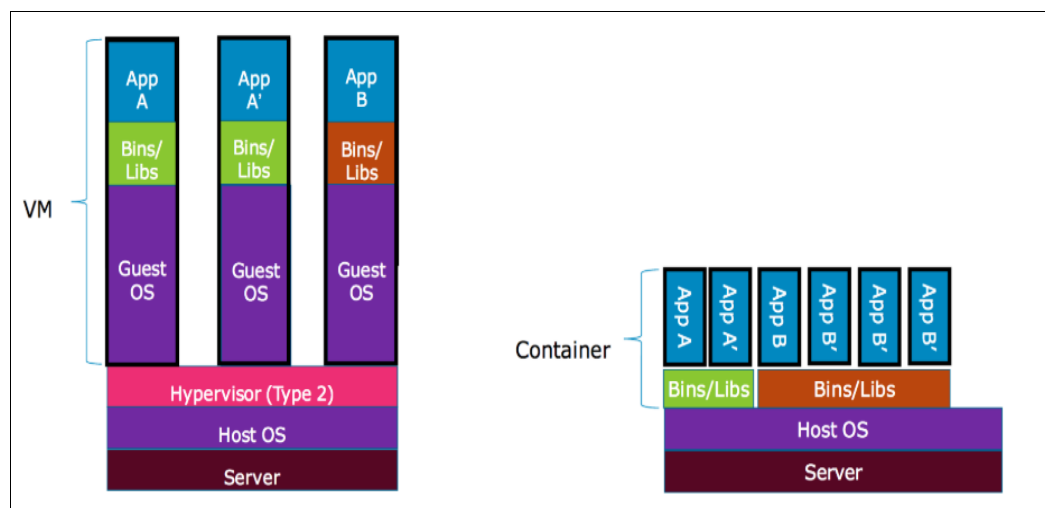


Figure 7-3 Virtual machines and Containers

7.3.3 Container key points

Consider the following points regarding containers:

- ▶ Containers are virtual software objects and include an application and all of the elements it needs to run.
- ▶ A container includes the benefits of resource isolation and allocation and runs in a host environment. They also share the services that are provided by the host.
- ▶ Containers also help you build high-quality applications faster because you start with software that is installed and configured.
- ▶ Containers are portable. Any platform with a container engine can run containers.
- ▶ Containers are easy to manage:
 - Container images are easy to share, download, and delete (especially with Docker registries)
 - Container instances are easy to create and delete
 - Each container instance is easy and fast to start and stop
- ▶ Containers provide “just enough” isolation:
 - More lightweight than virtual machines
 - Processes share the operating system kernel, but are segregated
- ▶ Containers use hardware more efficiently. They feature greater density than virtual machines (especially Docker containers, which can share layers).

7.3.4 Container orchestration

Although containers provide an effective way to spin up several applications and better use the underlying operating system resources, by default it does not provide a mechanism to manage, control, deploy, and scale a containerized infrastructure in a coordinated and organized way.

Figure 7-4 shows several containers that are distributed on top of an infrastructure. The management and control of these containers is done individually, which increases the complexity and creates more effort to maintain such a design.

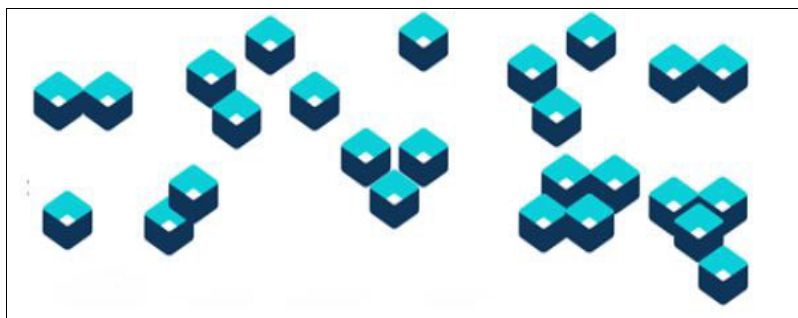


Figure 7-4 Application containers distributed in a non-coordinated way

To avoid such scenario, container orchestration technologies emerged.

In distributed computing, an *orchestrator* is the piece of software that is responsible for organizing and automating computing systems to ensure that these systems are run in a manageable and coordinated way.

As an analogy, think about a maestro role within an orchestra, whose job is to ensure that all musicians and instruments play under perfect harmony.

A container orchestrator is required to synchronize the communication, scheduling, configuration management, load balancing, and other important components within a containerized infrastructure, as shown in Figure 7-5.

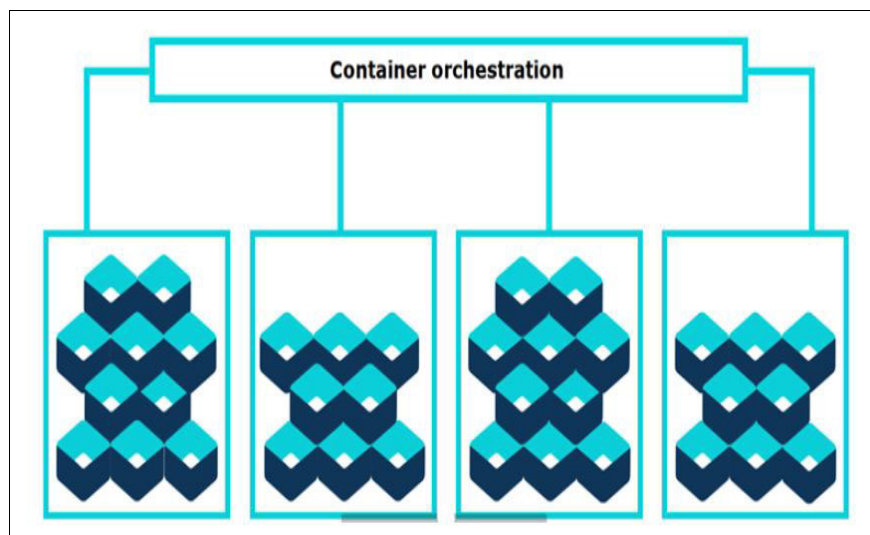


Figure 7-5 Container Orchestration allows for easier container management

Container orchestration is mainly responsible for managing the deployment, placement, and lifecycle of containers by using the following features:

- ▶ **Schedule management:** Distributes containers across nodes by using a scheduler.
- ▶ **Cluster management:** Federates multiple hosts and makes them part of a container cluster.
- ▶ **Service discovery:** Knows where containers are located and distributes client requests across them. For more information, see 7.2.2, “Service discovery” on page 199.
- ▶ **Replication:** The process of horizontally scaling containers across the cluster.
- ▶ **Health check:** Detects and replaces unhealthy containers and nodes.

Container orchestration also uses the following features that are critical on a cloud infrastructure:

- ▶ **Provisioning:** The ability to spin-up and down containers as necessary.
- ▶ **Monitoring:** Provides health checking mechanisms, such as container self-healing, overall node’s status, and performance metrics.
- ▶ **Configuration management:** Allows customization of several configuration parameters for containers and for the orchestrator by using APIs and configuration files.
- ▶ **Auto scaling:** Refers to the ability of auto-adjusting the number of containers according to the workload demand.
- ▶ **Networking:** Abstracts network-related tasks and provides the means to allow communication between containers to occur and mechanisms to expose their traffic externally as necessary.
- ▶ **Load balancing:** Distributes the incoming traffic over multiple containers within a cluster.

- **Policy Management:** The ability to customize several container-related policies, such as resource quotas and security related functions.

7.3.5 Kubernetes

Kubernetes is the leading orchestration technology for container management in use today. IBM supports the development of Kubernetes-based solutions and being among the top contributors to the open source project, plays a key role to its development. In that sense, the IBM LinuxONE platform can run and scale several Kubernetes clusters within its single hardware footprint. IBM strategy on supporting the Kubernetes development allows for organizations to take the most out of the IBM LinuxONE platform during their shifts to a modern cloud infrastructure.

Kubernetes automates several container activities and provides centralized infrastructure management for Container as a Service (CaaS) solutions. Kubernetes also supports further extending its capabilities by supporting the use of add-ons in such a way that the product is flexible to meet various demands.

The platform also includes a powerful REST API interface that allows organizations to further integrate their businesses with the orchestration solution. When the Kubernetes API is used, it is possible to query and retrieve cluster information and control cluster resources, which allows developers to take the most out of the platform. It also includes powerful CLI solutions that allow administrators to easily automate daily tasks.

Kubernetes is a feature-rich orchestrator and is a critical component for containers manageability. In that sense, it is composed of several components that form the core foundation towards pursuing a modern cloud infrastructure. Figure 7-6 shows a simplified Kubernetes architecture that is suited to run small to medium container workloads.

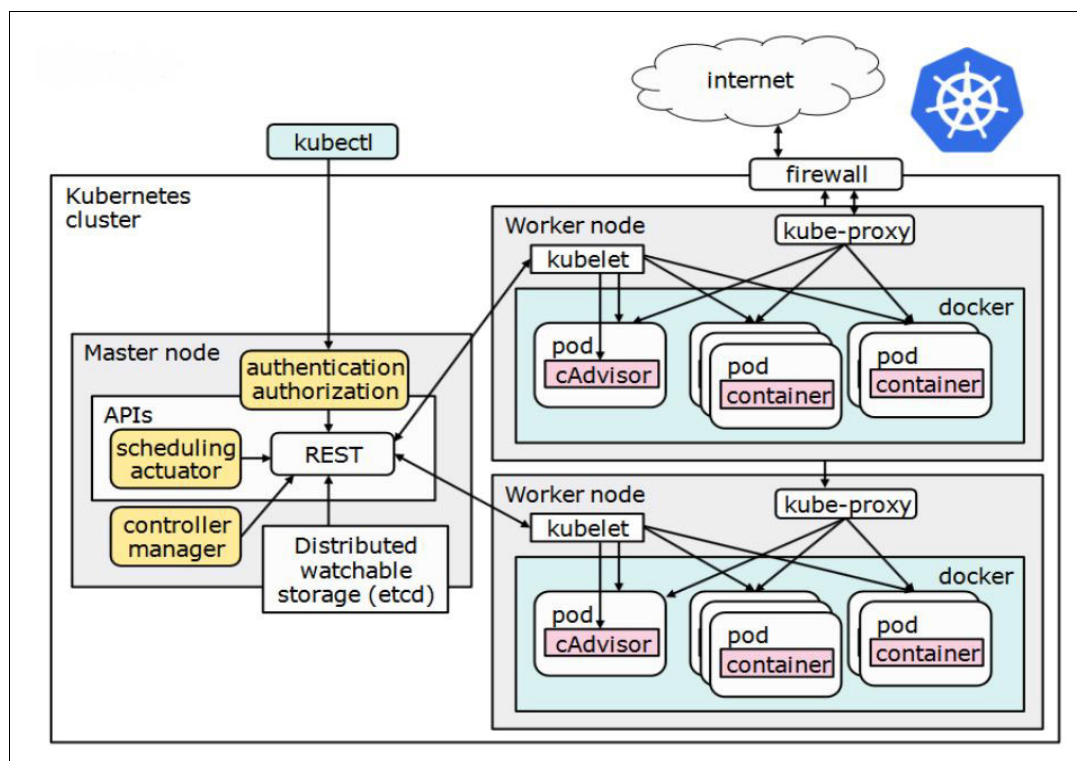


Figure 7-6 Kubernetes is a feature-rich container orchestration platform

Master node

The controlling services in a Kubernetes cluster are in the master node. These components operate as the main management contact points for administrators. They also provide many cluster-wide system management services for the worker nodes. Because of the criticality of the master node and its important role, it is recommended that it is redundant and made highly available.

Worker node

A worker node is a single physical server or virtual machine in which containers are deployed. The master node is responsible for controlling all worker nodes in such a way that interaction with worker nodes is rarely needed. Although it is possible to run a worker node within the same master node server, this implementation often is discouraged. Similarly, as with the master node, it is recommended that Kubernetes implementations count with at least two worker nodes for high availability.

Pod

A pod is the smallest unit within a Kubernetes cluster and its purpose is to run containers. Each pod has its own IP address and shares a PID, network, and hostname namespaces. By definition, containers are ephemeral in nature in such a way that pods must implement mechanisms. Storage persistency is not a default configuration.

Service

A service is a collection of pods that are available as an endpoint. Whenever a service is defined in a Kubernetes cluster, a single IP address and DNS name is assigned to it. By abstracting all of the routing and load balancing information, a service provides an effective way for external clients to access its respective pod collection.

Figure 7-7 shows how defined services in a Kubernetes cluster can be used to allow external clients to communicate with a collection of pods.

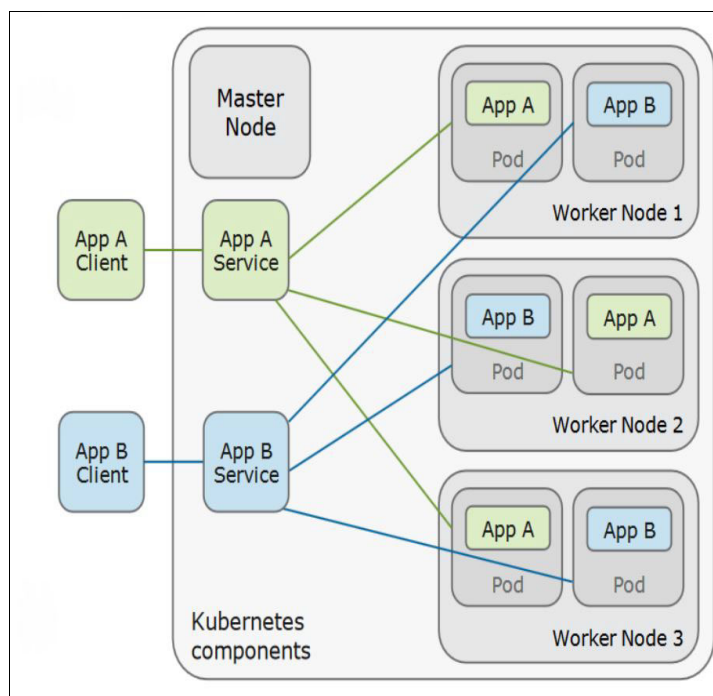


Figure 7-7 Kubernetes abstracts all the routing and load balancing across the cluster

7.3.6 Security in Kubernetes

Most emerging technologies that quickly gain adoption, such as Kubernetes, rapidly become a threat for malicious individuals and software that attempt to break into and steal sensitive data. As with any critical cloud component, Kubernetes clusters must also be secure and its implementation always be done with a security mindset.

One of the most important required security practices is to ensure that the Kubernetes cluster is always up-to-date and free of known security vulnerabilities. This requirement also is extended to containers that are running within it. In that sense, it is important to include strictly defined security controls to ensure that container images are always updated to releases that do not introduce any security exposure to the environment.

Next, we describe several important Kubernetes cluster components.

Kubernetes API service and etcd data store

The Kubernetes API service and etcd are the main components that require special attention in the Master node. The API provides mechanisms to control the whole cluster definitions. If unintended access is granted to the Kubernetes API, control of the entire cluster management is taken.

By default, Kubernetes requires every request to go through the following stages before access to the API server is granted:

- ▶ **Authentication:** Validates the identity of a registered user or service account.
- ▶ **Authorization:** Limits the permissions of authenticated users and service accounts to ensure that they can access and operate only the cluster components to which they have permission.
- ▶ **Admission control:** Validates or mutates requests before they are processed by the Kubernetes API server. Many Kubernetes features require admission controllers to properly function.

Figure 7-8 on page 209 shows the default cluster security settings that address the authentication, authorization, and admission control layers. It also adds a VPN server and client as an extra security layer measure to secure the connectivity between the Kubernetes master and worker nodes.

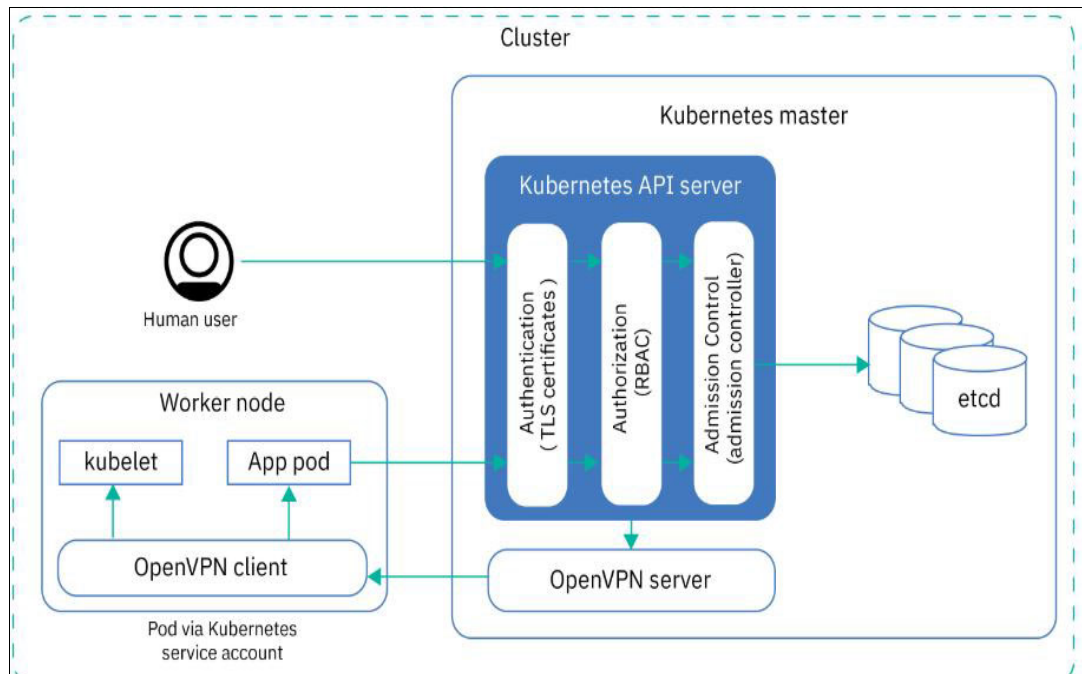


Figure 7-8 Kubernetes cluster security settings

The following security measures can be implemented for securing the Kubernetes API server on IBM LinuxONE:

- ▶ Master node hardening by using Linux traditional tools (SELinux, Two-factor authentication, and VPNs)
- ▶ Encryption of all data at-rest and in-flight (Pervasive encryption)
- ▶ Secure communication by using TLS
- ▶ OpenVPN connectivity to worker nodes for authentication control
- ▶ Fine-grained access control
- ▶ Admission controllers

The etcd data store that is in Kubernetes is responsible for storing all the cluster definitions, including application *Secrets*. Because of the importance of such component, it is important to ensure that its data is properly encrypted. When implementing a backup policy for etcd, it is also important that the backup is stored on encrypted disks or tapes and that its private keys are safely guarded.

The Kubernetes master represents the most important component within the cluster. When using Kubernetes in a private cloud environment, it is important to ensure that it does not expose any management ports externally, in such a way that the overall cluster management is possible only from within an on-premises infrastructure. Similarly, when deploying hybrid cloud environments that require access to cluster components, the use of VPNs for authentication and using mutual authentication and key rotation controls are highly encouraged practices to be adopted by organizations.

Worker nodes

The Master node is responsible for controlling and managing the entire cluster definitions and ensuring that the cluster resources are in a healthy state. The Worker nodes are the components that are responsible for running applications within the Kubernetes cluster. That is, these nodes are the components that perform the real work.

Unauthorized access to worker nodes can have disastrous effects on an organization's reputation and therefore, require the implementation of proper security controls. Figure 7-9 shows an overview of required security measures to consider when creating and managing worker nodes.

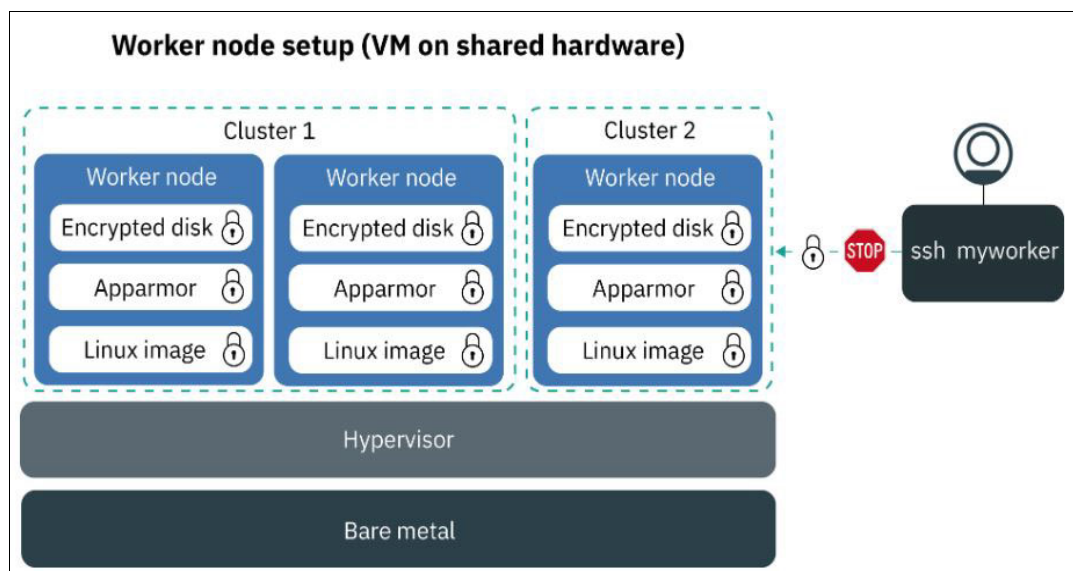


Figure 7-9 Proper security controls in Worker nodes ensures that applications are safe

The following security features must be considered when securing a Kubernetes worker node. Many of these controls can also be applied to master node security as required by your security policy:

- ▶ **Linux images:** Kubernetes worker nodes should be used only for running your *containerized applications*. In that sense, apart from implementing required programs as required by the governing security policy in place, the general guidance is to keep the Linux image as simple and minimalist as possible. Such approach greatly reduces potential attacks.
- ▶ **Compute isolation:** Kubernetes provides basic network security controls that allow an administrator to define cluster communication policies. For enhanced control, consider the use of host-based or network-based firewalls that deny all incoming and outgoing traffic by default. Then, open only the bare minimum required traffic as required for the business to operate.
- ▶ **Dedicated systems:** In general, because the worker node should be used for spinning up and down containers only, each node must be dedicated to a single virtual machine.
- ▶ **Encrypted disks:** As with the master node, it is recommended that all storage volumes that are used by containers use the Pervasive Encryption feature of IBM LinuxONE for encryption of data at-rest. This IBM exclusive feature ensures that your application data is held securely, even against physical intrusion attacks.

- ▶ SELinux/AppArmor policies: SELinux and AppArmor policies provide fine grained access control to operating system resources to prevent malicious activities to be performed on the target system without the proper privileges. Similarly, ensuring that the policies cannot be modified by general users is a recommended security practice and helps to guarantee the overall cluster security consistency.
- ▶ Disable SSH access: Remote access to worker nodes can present a security concern. As described in “Worker node” on page 207, the overall cluster management is done only by using the master node. Apart from typical maintenance activities, such as operating system patching and persistent storage management, the worker nodes are required to operate independently of human interactions.

In general, access to the server’s console is limited by using Linux traditional TTYs (Teletypewriter terminals) instead of pseudo terminals. Whenever disabling SSH is not feasible, ensure to implement proper security controls, such as privilege revalidation and the use of Linux `tcp_wrappers` to ensure that it accepts only incoming traffic from intended clients.

Networking

To protect your network and limit the range of damage that a malicious attacker can inflict, make sure that workloads are as isolated as possible and that the number of apps and worker nodes that are publicly exposed is limited.

The use of VLANs to logically subdivide a physical network is a good security measure even for modern cloud environments. Consider the placement of the worker nodes on VLANs that are separated from other components of your organization, such as heritage applications, and allow the communication between those only when strictly necessary. It is also possible to further extend this requirement on placing the master node in more separated VLANs to ensure that management access to the overall Kubernetes cluster is completely isolated, controlled, and easily auditable.

Networking considerations should also be in place to meet an application’s regulation requirements. A good security practice is to host sensitive processing applications and databases on separated networks, clusters, and network zones, than general, less restricted, applications.

Consider the following security features while securing the network from external attacks:

- ▶ Limit the number of publicly exposed applications
- ▶ Keep worker nodes private and as isolated as possible
- ▶ Limit public internet connectivity

Persistent storage

Whenever provisioning persistent storage, ensure that the backend devices are properly encrypted. The IBM LinuxONE platform provides Pervasive Encryption, which ensures that the application data is secured even from unauthorized physical access. Similarly, consider safe guarding encryption keys to further prevent unauthorized data access to sensitive information.

Monitoring and logging

The key to detect malicious attacks in Kubernetes is the proper monitoring and logging of metrics and events that occur across the cluster. Monitoring and logging also have the included benefit of helping administrators to better understand the cluster capacity and resource availability for applications. This knowledge is used in planning current and future growth demands and protecting applications from downtime.

Container images and registry

Every deployment is based on an image that holds the instructions for how to spin up the container that runs an application. These instructions include the base operating system inside the container and extra components that are required to run the application. For enhanced application security, the base image must be protected and mechanisms to check the image's integrity and consistency are required.

To protect applications, consider addressing the following areas:

- ▶ Automate the image build process by integrating it into the CI/CD pipeline and limit access to the number of people in the organization that can access the build process
- ▶ During the build process, ensure that all components are updated to their latest versions to reduce the risk of introducing security vulnerabilities into the environment
- ▶ Scan and healthcheck images before they are deployed into production. Consider adopting automation mechanisms for threat detection and application sanity.
- ▶ Regularly scan running containers and redeploy them with updated images whenever a new vulnerability is introduced to its backing components.
- ▶ Adopt mutual authentication mechanisms for cross application communication.
- ▶ Avoid the use of non-encrypted protocols to avoid man-in-the-middle (MITM) attacks and network spoofing that are intended to leak information.

Workspace and Container isolation

Kubernetes namespaces are a way to virtually partition a cluster, provide isolation for deployments, and limit the users that are allowed interact with it. With namespaces, it is possible to organize resources across worker nodes and across zones in multizone clusters.

The following security features can be considered as you are securing your containers:

- ▶ Limit the number of privileged containers. In general, avoid running containers with root permissions. Never provide means to application containers to interact with the Docker daemon, such as allowing communications by using the TCP socket or bind mounting the Docker socket file (`/var/run/docker.sock`). These use cases are discouraged and can cause security exposure if unintended access is gained to such containers.
- ▶ Set CPU and memory limits for containers. This limit prohibits misbehaving applications from causing resource starvation on the cluster and affecting other applications.
- ▶ Apply operating system security settings to pods. By enabling the `securityContext` feature on the pod specification, several host-based security definitions can be applied to the container, which greatly limits its capabilities. For example, SELinux (for fine-grained access control) and seccomp controls (for system calls filtering) can be applied.

Personal information handling

Kubernetes provides different mechanisms for ensuring the security of personal information in its resources and container images. Among the strategies that can be applied to safeguard the storage of privileged information and handling of intellectual property, the following methods are recommended:

- ▶ Use Kubernetes secrets to store personal information. Secrets provide a convenient way to store sensitive data and avoid accidental exposure of information. With secrets, containers do not need to hardcode any confidential information within its code.
- ▶ Store personal information only in Kubernetes resources that are designed to hold personal information. For example, a good security practice involves not referring to any identifiable information of an individual in the name of a Kubernetes namespace, deployment, service, or config map.

- Use a Kubernetes `imagePullSecret` to store image registry credentials. Similarly, because personal information must be safeguarded, the `imagePullSecret` feature that is on Kubernetes also allows for safely storing and retrieving registry credentials in such a way that only authorized personnel with access to the secret can manipulate the registry.

In general, do not store personal information in container images or registry namespaces. For proper protection and encryption, store registry credentials in Kubernetes `imagePullSecrets` and other personal information in Kubernetes secrets. Because images are composed of several layers, as described in “Images and containers” on page 202, deleting an image might not be sufficient to completely remove this personal information if personal information is stored in a previous layer of an image.

7.4 Containers management at scale

This section provides an introduction to how IBM LinuxONE can meet the scalability and performance demands when orchestrating containers by using Kubernetes. It also introduces different deployment strategies that are commonly used for microservices and containers.

7.4.1 IBM LinuxONE as the container platform

The union of containers and Kubernetes-based technologies represents a huge shift on the way that cloud-ready and traditional applications can be modernized, scaled, and deployed to satisfy the fast changing requirements that were introduced by the present digital era. With the growing adoption of on-premises environments and containers, the requirements for a reliable and agile infrastructure are, once again, going to be the determinant factor for the success of an organization.

One of the common problems that typically occur during the deployment of a container-based infrastructure lies on hardware and architectural limits. The choice of the right architecture is important because several requirements are imposed during the container orchestration process.

Also, organizations want to simplify operations, meet compliance requirements, ensure overall system-wide security, and reduce costs. In that sense, the IBM LinuxONE platform is the ideal choice for running Kubernetes, containers, and traditional Linux-based workloads in a single hardware footprint.

A common scenario that is seen in Kubernetes implementations in on-premises infrastructures goes back to a classic problem that is commonly seen during the pre-virtualization era: server farming². This issue occurs because although containers are typically lightweight components, great processing power is required to orchestrate, scale, and manage an enterprise infrastructure. In that sense, it is important to consider an organization's needs and plan for future growth.

Implementing a container-based infrastructure comes with its own challenges. Moreover, the effects of server farms can be disastrous for a Kubernetes cluster and its applications. Server farming can cause the following adverse effects:

- Higher complexity: The overall infrastructure is much more complex to build and manage because several servers are spread across the data center

² In this context, *server farms* refers to an increased management complexity that was introduced by having several distributed systems spread across an infrastructure because their capacity alone cannot meet the required demand.

- **Increased latency:** The communication across the cluster components takes longer, which can introduce unnecessary wait times for users.
- **Limited scalability:** Kubernetes nodes typically run on x86 hardware that is not suited to meet the scalability demands required by many organizations. A cascading effect occurs; that is, scale more, more servers are needed.
- **Low performance:** Cluster nodes can easily have all its resources starved and by affecting other applications, cause unplanned downtime.

The IBM LinuxONE platform is ready to meet all the demands for building an effective and efficient Kubernetes infrastructure. As described in 7.2.1, “Microservice architecture” on page 197, the IBM LinuxONE Emperor II can scale up to 2 million Docker containers in a single system without affecting performance impact. Therefore, it offers unmatched microservices scalability then alternative x86 variants. Also, Emperor II can also move data faster than alternative platforms with a 2.1x higher data processing throughput.

IBM LinuxONE Emperor II can also serve up to 30 billion web data requests a day, which ensures that the platform can meet current and future analytics, social media, and mobile demands, which are platforms that often include many user interactions per day.

The LinuxONE platform is built to run at processor utilization rates as high as 100 percent. This rate allows for microservices and containers to take the most of the platform. Moreover, features, such as Pervasive Encryption, LPAR workload isolation, IBM Secure Service Container (SSC), and Live Guest Relocation positions IBM LinuxONE as the most secure and highly available platform in the industry.

Kubernetes provides the tools that allow a container-oriented infrastructure to exist, and IBM LinuxONE provides the ground foundation that allows it to occur. When building a container on-premises infrastructure, organizations find in the LinuxONE platform all of the necessary capabilities to transform their businesses and securely modernize their applications.

7.4.2 Deployment strategies

Several strategies can be implemented during the deployment of microservices in a Kubernetes cluster. Kubernetes automates and abstracts several components for scaling an application and ensuring it is available during its roll-out. Some of these deployment strategies are described next.

Re-create with downtime

This deployment strategy is the most common and directly contrasts with the way traditional applications are handled. Re-creating with downtime involves stopping all required application components and then spinning-up the new version later. This pattern involves a downtime for the application between the period on which the older version is being stopped and the new release is being deployed.

For non-mission critical applications, this downtime can be acceptable. This strategy is also the most simple deployment pattern that can be implemented.

Rolling updates

Rolling update refers to a deployment strategy that achieves zero downtime. One of the requirements of such approach is that the application to be updated must be scaled to at least two replicas or more.

Rolling updates individually replace each container with the new version of an application. It can achieve zero downtime because Kubernetes automatically load balances users only to running replicas of the application.

Depending on the application requirements, it is also possible to instruct Kubernetes to probe the application and ensure it is fully started before routing users to the new version. Such a scenario can easily be achieved with Kubernetes readiness probes that allow developers to specify sanity checks with which the cluster can identify when an application is ready for servicing. Finally, after the new version is successfully rolled-out, Kubernetes stops and remove the previous containers.

The drawback of rolling updates is that it requires more computing resources during the roll-out period. Because Kubernetes spins up the new version side by side with the older running version, rolling updates can present a problem for resource-intensive containers.

The benefits of such approach are zero downtime, reduced impact, and easy rollback mechanisms if a need arises to return to the previous version. As with updating, the rollback can also be done without any downtime.

Blue/green deployments

Blue/green refers to the deployment of a new version of an application (green) that is run simultaneously alongside its previous version (blue), in such a way that both versions coexist within the cluster. The biggest difference from rolling updates is that no traffic is routed to the new version.

The green deployment is then tested and, if the result is satisfactory, the traffic is routed to it and no longer to the blue (old) application. If the deployment is unsuccessful or presents problems, the new version of the application is shut down and the older application continues to operate. If problems are discovered later, the switch to the green (new) version occurred, and the rollback can be done by routing the traffic back to the blue (old) version.

The drawback of such approach is that it causes a resource overhead because two duplicated environments are running simultaneously.

The main benefits of this approach include zero downtime and the possibility to perform tests before routing users to the new updated version. By having two environments running simultaneously, the rollback to the older version is quick and greatly simplified.

Canary deployments

Canaries are similar to blue/green deployments in nature. However, instead of routing all users directly to the new version, the shift occurs gradually.

The basic idea behind canary deployments is to prevent a major business impact if the new updates are not well-accepted or start presenting problems. Another benefit of canaries is the possibility to extract metrics that are based on user feedback in such a way that developers can monitor the interaction with the application from a small set of users to determine whether the update was well-accepted. After the update is ready to be rolled-out to all users, all traffic is routed to the new version.

In Kubernetes, the roll-out of canary deployments is done in a controlled way. First, one replica that includes the new application is deployed. After the developers are comfortable with the change, these replicas are scaled up to match the same number of replicas as exist on the old version. Finally, the old version is removed from the environment.

A/B testing

A/B testing allows organizations to make business-oriented decisions according to metrics and user acceptance. This deployment relies on the concept of canaries, on which the changes initially target a small subset of users. When Kubernetes is used along with Istio, the traffic can be split among microservices releases by defining different weight values to each version.

Summary

The following benefits can be realized by using microservices, Kubernetes, and the IBM LinuxONE platform together from an organizational perspective:

- ▶ Improved business agility: Deployment time and speed to market are dramatically reduced, which results in huge gains in business efficiency.
- ▶ Improved business predictability: Kubernetes improves CI/CD pipelines and the automation of several deployment strategies. In that sense, canary deployments and A/B testing mechanisms helps organizations to better drive their businesses according to user's acceptance.
- ▶ Improved developer productivity: The IBM LinuxONE platform can scale several isolated environments on a single hardware footprint. With multiple Kubernetes clusters, it allows for hosting isolated testing, development, and production environments with optimal performance and unmatched security by any other available platform.
- ▶ Improved operational efficiency and costs: The adoption of an agile infrastructure with the flexibility of open source components allow for businesses to save on software licenses while dramatically improving their efficiency.

7.5 IBM Cloud Private overview

IBM Cloud Private is an application platform that is used to securely develop and manage on-premises, containerized applications. The integrated container's management environment includes Kubernetes, a private image repository, a management console, and monitoring frameworks.

IBM's approach in adopting the Kubernetes platform as an open source-based container management system targets operations and development teams. With containerization, developers can treat configuration as code to enable and use a DevOps toolchain.

By using IBM Cloud Private, development and operations teams share a flexible cloud environment behind their firewalls to create microservices-based applications, modernize applications by using cloud-enabled middleware, and securely integrate between the two. IBM Cloud Private allows customers to have public Cloud capabilities within their own managed and controlled infrastructure.

The union of IBM Cloud Private with IBM LinuxONE allows for customers to further extend their cloud capabilities by securely combining the rapid deployment speed that is required by most modern organizations with the built-in hardware security capabilities of the platform, such as complete LPAR isolation (EAL5+ certification), and Pervasive Encryption of data at-rest and in-flight. It also enables the use of LinuxONE's exclusive capabilities, such as IBM Secure Services Containers (SSC), with IBM Secure Services Container for IBM Cloud Private.

7.5.1 Key aspects

IBM Cloud Private delivers the following key aspects that are required for any enterprise cloud strategy:

- ▶ Elastic runtimes for enterprise developers with which they can use modern architecture and techniques on the cloud.
- ▶ Cloud-ready IBM middleware to use investments in IBM processes and tools on top of the same Kubernetes environment.
- ▶ Production-ready operations so that you can use your toolchain inside of your data center or use the built-in support for identity and access management (IAM), logging, and monitoring.
- ▶ Built-in support for continuous delivery so that the developers become productive on the platform.

7.5.2 IBM Cloud Private architecture

In the private cloud market, the initial focus is shifting from infrastructure as a service (IaaS) to containerization. This shift comes to aid the overall productivity of developers and to enable DevOps.

The private cloud architecture (see Figure 7-10) provides container-as-a-service (CaaS) and platform-as-a-service (PaaS) enablement for private workloads. Often, a containerized platform is based on container-based orchestration, such as Kubernetes.

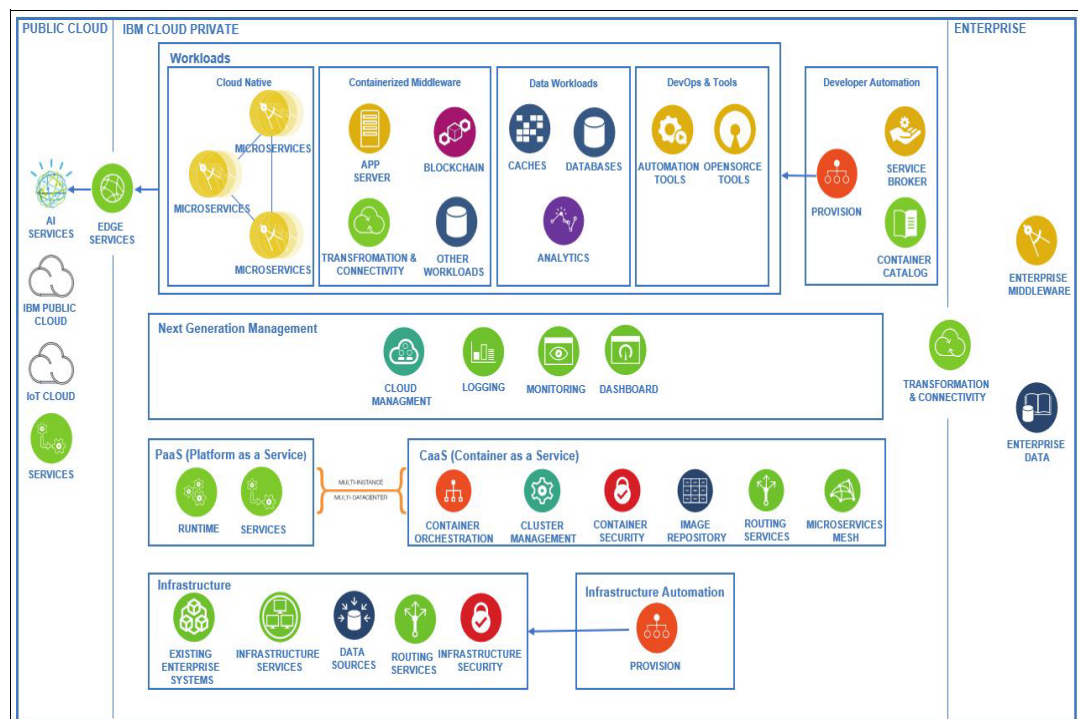


Figure 7-10 Private cloud reference architecture

The private cloud architecture provides several benefits, as shown in Figure 7-10 on page 217:

- ▶ Container orchestration, which is at the core of the architecture. This layer provides cluster management, security capabilities, image repositories, routing services, and microservices mesh.
- ▶ A PaaS layer, which can enhance a container environment by providing higher-level runtimes and service bindings that allow for an easier development experience.
- ▶ The CaaS and PaaS layer, which sit over an infrastructure layer that provides compute through virtual machines, network, storage, and security.
- ▶ Automation and orchestration for the underlying infrastructure, which allows for an infrastructure-neutral software-defined data center. Infrastructure automation can provide predefined infrastructure templates to create repeatable patterns.
- ▶ A private cloud platform provides monitoring for container-based applications to provide logging, dashboards, and automation. It supports network and storage policy-based controls for application isolation and security, and automated application health checking and recovery from failures.
- ▶ The ability to run isolated containerized workloads for several patterns, such as cloud-native, data workloads, integration workloads, tool workloads, and some middleware, such as Java Application Server.
- ▶ A catalog of workloads that you can provision over containers to improve developer experience.
- ▶ Enhanced data control and sovereignty and compliance with an enterprise own security and governance policies

7.5.3 IBM Cloud Private Security

IBM Cloud Private allows customers to deploy an on-premises Cloud-based infrastructure. This approach includes the following benefits that help to avoid most of the concerns of deploying infrastructure on top of traditional public clouds:

- ▶ Infrastructure Physical security
- ▶ Identity and access management (IAM)
- ▶ Logging and auditing
- ▶ Data ownership and isolation
- ▶ Externally exposed APIs

With IBM Cloud Private, the infrastructure control is returned to customers so that they can scale up their cloud portfolio within their own policy and regulation requirements. By implementing authentication and authorization policies to organizational requirements, customers can further use IBM Cloud Private to ensure that only authorized personnel can access their cloud infrastructure management.

Being built on top of IBM Security engineering practices and open security standards, IBM Cloud Private also ensures that the deployed on-premises cloud is protected from external attacks and internal threats. For example, IBM conducts penetration testing for every IBM Cloud Private release. This testing ensures that customer's on-premises infrastructure is inherently secure.

Another growing concern of public clouds as opposed to private clouds concerns data management. With IBM Cloud Private, all data is owned, managed, and controlled by the data owner. This feature is important for customers that handle sensitive or privileged data.

Because IBM Cloud Private is built with a security mindset, all application management interfaces and data handling are accessible from within only an organization security zone in accordance with the organizational requirements in place.

Because IBM Cloud Private includes a portfolio of Cloud-ready applications, the deployment of such an infrastructure is more secure than traditional on-premises infrastructure. This occurs because by using IBM Cloud Private, customers can apply uniform security policies across all applications within its simple management interface instead of traditional on-premises infrastructures that rely on manual updates to each service on top of a stack of applications.

Application Security

Vulnerability Advisor is a feature of Cloud native and Enterprise editions of IBM Cloud Private to retrieve security status for container images on top of the IBM Cloud Private registry. It also checks for the security and compliance status of running containers that are deployed within an infrastructure. Vulnerability Advisor can also be configured to scan private registries to ensure that even customer-owned images are scanned for potential security threats.

Such a feature allows for customers to quickly assess and address possible security threats and act on their remediation quickly. It is also possible to review security reports with ease directly from the IBM Cloud Private management interface.

Vulnerability Advisor security notices are reviewed, processed, and made available by IBM Security to ensure that customers receive potential threat reports in time about their managed infrastructure, which greatly reduces potential intrusion threats with ease.

Note: For more details on Vulnerability Advisor, see [Managing image security with Vulnerability Advisor](#)

Service IDs and API Keys

One of the growing concerns of cloud applications and microservices-based workloads is how each component of an application is communicating with each other and how secure are such credentials that are propagated within the network. Moreover, another important factor to consider is how services or applications that are outside of the on-premises cloud infrastructure can interact with it.

IBM Cloud Private allows for developers to further use their cloud applications by using Service IDs and API keys. Service IDs identify an application or service; API keys are used as an authentication method for such Service IDs.

By creating specific Service IDs and their respective API keys, developers can use credentials that use the principle of least privilege. That is, it allows connecting applications to be granted access only to the minimum set of information that is required for its legitimate purpose and functioning.

Service IDs are not tied to a particular user in such a way that if a developer leaves the organization, the service ID remains, which ensures that the application or service in question continues to operate.

Also, by creating individual credentials for each service, if an API key is compromised, it does not give access to other resources across an infrastructure. IBM Cloud Private allows for the quick replacement of lost API keys across its infrastructure in such a way that credentials can be quickly refreshed by using the IBM Cloud Private management interface.

Authentication and authorization

IBM Cloud Private supports the following authentication protocols for users:

- ▶ OpenID Connect (OIDC) based authentication is provided by IBM Cloud Private through the WebSphere® Liberty Server. This protocol is backed by a WebSphere Liberty-based OIDC server for providing local and LDAP directory-based authentication.
- ▶ Security Assertion Markup Language (SAML) based federated authentication, where IBM Cloud Private can be configured to use it from an enterprise SAML server.

OIDC and SAML integration mechanisms that are available on IBM Cloud Private allow for customers to integrate their authentication mechanisms that are in use within their infrastructure. By doing so, organizations can effectively control and revoke access as needed to all of its corporate infrastructure in a centralized way if a security threat arises from the inside.

As it should be adopted by most organizations, the use of centralized authentication mechanisms is highly recommended. The use of local credentials within IBM Cloud Private is discouraged and such access available to trusted and privileged personnel within your organization.

The authorization mechanism in IBM Cloud Platform is available by using role-based access control (RBAC). This feature allows for cluster administrators to grant and revoke privileges directly from the IBM Cloud Private management interface to individuals according to their business needs. For example, users with the role of Auditor have only limited access to support their business role.

Another key feature of RBAC within IBM Cloud Private is assigning privileges to specific Kubernetes namespaces in such a way that members of a team can manage only resources that pertain to the namespaces that were granted privileges to them. Such a feature ensures that individuals access only the resources that they are intended to, which provides namespace manageability isolation and enhanced security for your CaaS infrastructure.

Note: For more information about role-based access control, see [IBM Knowledge Center](#).

7.5.4 IBM Cloud Private features

IBM Cloud Private can be managed in various ways. The default interface that is provided within IBM Cloud Private can manage, monitor, and troubleshoot applications and clusters from its single, centralized, and secure management console. Figure 7-11 on page 221 shows how easy and intuitive managing an on-premises cloud infrastructure is possible with IBM Cloud Private.

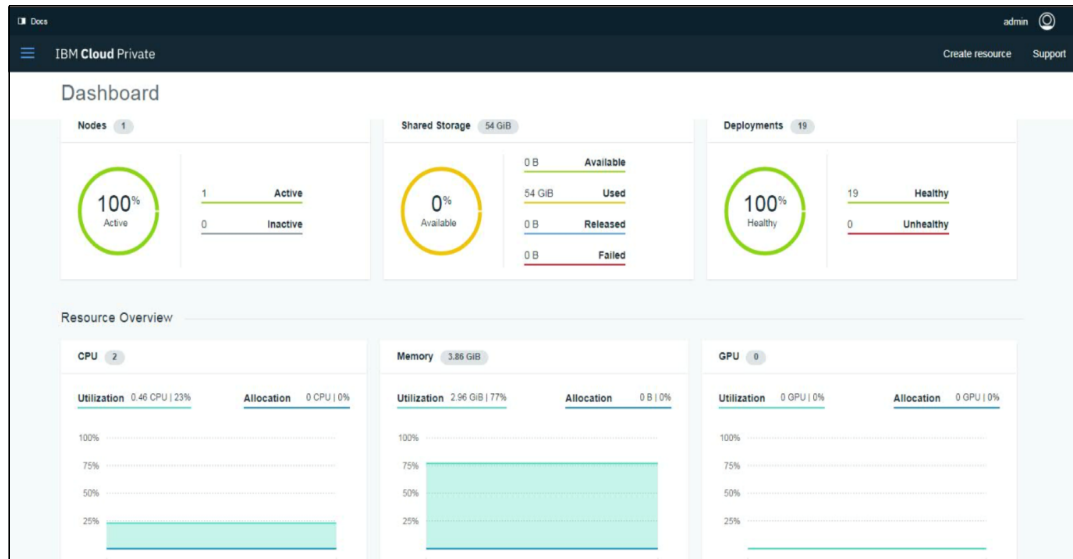


Figure 7-11 IBM Cloud Private management console

Developers and systems administrators also can use the IBM Cloud Private Command Line Interface (CLI). This CLI automates tasks, retrieves cluster and application information, and integrates it with other business processes as required by the governing policy.

Note: For more information about IBM Cloud Private CLI installation and capabilities, see to Appendix A of *IBM Cloud Private System's Administrator Guide*, [SG24-8440](#).

Private Docker image registry

Integrates with the Docker registry V2 API to provide a local registry service that functions in the same way as the cloud-based registry service, Docker Hub. This local registry includes all of the same features as Docker Hub, but also allows you to restrict which users can view or pull images.

With IBM Cloud Private, authorized users can further extend their cloud application portfolio by pushing their custom images to the IBM Cloud Private private registry. Such images can then be used by members of the same namespace to deploy the required containers within the cluster. They also can be shared among several or all namespaces as required by the governing policy.

Application catalog

This feature provides a centralized location on which it is possible to browse for and deploy application containers within a cluster. With IBM Cloud Private, the catalog is composed of both IBM and third-party content.

Packages for other IBM products are available from curated repositories that are included in the default IBM Cloud Private repository list. The environment must be connected to the internet to update and retrieve the packages present in the catalog.

Figure 7-12 shows an overview of the services present in the IBM Cloud Private catalog, which include various technologies, such as AI, IBM Blockchain, DevOps tools, IOT, and Security.

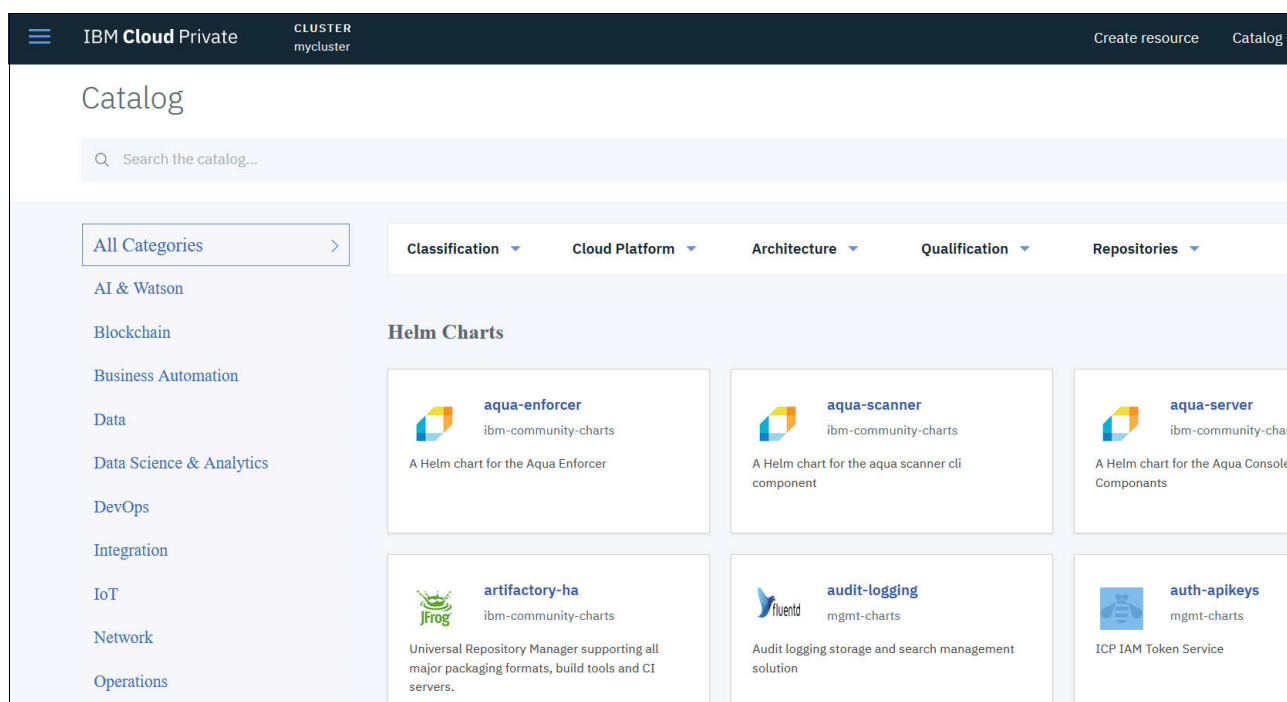


Figure 7-12 IBM Cloud Private Catalog

Isolated tenant networks

Calico enables networking and network policy in Kubernetes clusters across the cloud. It combines flexible networking capabilities with run anywhere security enforcement, which provides performances that are similar to a native kernel and enabling real cloud-native scalability. Calico is implemented without encapsulation or overlays, which provides high-performance networking. It also provides a network security policy for Kubernetes pods through its distributed firewall.

Because Calico uses the Linux Kernel's forwarding and access control capabilities, it provides a high-performance solution without more resources for encapsulation and decapsulation. Also, by interacting directly with the Linux Kernel, it avoids potential intrusion points because all security policies are managed directly from within the Linux stack.

The use of Calico to create isolated subnets for each project to add security during data transmissions and to reduce the chances of compromising applications and their data is encouraged for customers who are deploying IBM Cloud Private solutions across their infrastructure.

Robust monitoring and logging with ELK stack

Logging and monitoring are critical for Cloud workloads. Having a centralized way for operations and development teams to query for cluster status is important for most organizations. With the growth of a Cloud infrastructure, it is important that administrators can collect audit, performance, and logging metrics from their production workloads.

The Elasticsearch, Logstash, and Kibana (ELK) stack is a suite of open source tools that provide extensive log capture, retention, visualization, and query support for application log data. ELK also is the primary method for users to interact with their application log data.

IBM Cloud Private uses such open source tools to provide customers with a centralized logging facility with ease.

This process provides a centralized store for all logs and metrics, better performance, and increased stability when you access and query for logs and metrics. You can use the results from these queries to produce insightful graphs and reports with the IBM Cloud Private dashboard.

Such an approach shines when compared to the traditional way of handling a traditional IT infrastructure. That is, instead of managing logs and metrics from various sources, customers can rely on a single centralized resource to provide insights for their entire cloud infrastructure by using IBM Cloud Private.

IBM Cloud Private uses the following components to collect, store, and query logs and metrics:

- ▶ Elasticsearch
- ▶ Logstash
- ▶ Kibana
- ▶ Filebeat
- ▶ Heapster

Summary

IBM Cloud Private is a full featured product that can meet the growing demands of an on-premises cloud infrastructure. By using IBM Cloud Private, customers can securely manage and scale their cloud services with ease. This ability addresses the required speed to market demands and ensures compliance with the IT governing policies in place.

IBM Cloud Private meets all requirements for a modern agile infrastructure. By combining several secure capabilities, such as auditing, network policies and firewalls, role-based access control, and TLS encryption, the platform security is unmatched by other competitors on the private cloud market.

Note: For more information about IBM Cloud Private security, see Chapter 6 of *IBM Cloud Private System Administrator's Guide*, [SG24-8440](#).

7.6 IBM Cloud Private on LinuxONE

IBM Cloud Private security capabilities can be extended by deploying an on-premises infrastructure on top of the IBM LinuxONE platform. This section provides an overview of how the union of IBM Cloud Private with the LinuxONE platform can provide customers with an ideal infrastructure for performance and security to achieve unmatched speed to market and compliance with regulatory requirements.

7.6.1 Security levels for containerized applications on LinuxONE

Several layers of security can be applied for deployed containers on top of IBM Cloud Private for LinuxONE. Each of these layers is equally important. Thorough planning must be considered to ensure that the entire cloud infrastructure is kept secure in agreement with the governing policy that is in place.

Although security best practices suggest the duties of administrators be separated to reduce the risk of cybersecurity threats, they are often integrated because of business practices. As a security strategy is developed, consider that the different administrator roles and responsibilities are paramount and might require the realignment of your organization's security policies and processes. A separation of duties limits who can access and manage the security mechanisms of your IBM Cloud Private infrastructure and IBM LinuxONE platform.

Ideally, when rolling out a separation of duties policy within your organization, the following aspects must be considered:

- ▶ Access and management to Cloud resources and application workloads must follow the principle of least privilege.
- ▶ Auditors must query and audit how, when, and who accessed or modified a determined resource in a specific time in such a way that the cloud is kept secure in accordance with the corporate policy.

The following levels of security are applied:

- ▶ LinuxONE Hardware
- ▶ Private Cloud Security
- ▶ Containers and Orchestrator (Kubernetes)
- ▶ Applications and Microservices

These security levels are described next.

LinuxONE hardware security

LinuxONE provides customers with a combination of a highly scalable standards-based platform with the highest level of security at the core. Security is built in at the lowest levels of the platform for LinuxONE.

The most important technologies for ensuring a high level of protection are Pervasive Encryption, the Hardware Security Module (HSM), and IBM Secure Service Container.

Pervasive encryption

Pervasive encryption is an IBM exclusive feature that is present on the IBM LinuxONE platform that provides encryption at-rest and in-flight. Encryption of data at-rest occurs directly at the Linux volume level and therefore, it is handled transparently and does not require application code changes.

This approach enables companies to encrypt all of their data by default with little compute overhead. One of the benefits of the LinuxONE platform is the extent of the security services. Because the IBM LinuxONE architecture is built with a security mindset, it includes security capabilities that are integrated at every level of its hardware and software stack.

LinuxONE-based security is designed to encrypt data in bulk. Therefore, it is possible to encrypt all the data that is associated with an application or a database at one time.

IBM Cloud Private for LinuxONE can use Pervasive Encryption features. By creating Linux Unified Key Setup (LUKS) encrypted volumes and managing them by using the dm-crypt interface, containers can be deployed automatically with data at-rest encryption.

Note: It is recommended that encrypted volumes for your IBM Cloud Private infrastructure be configured *before* the product is installed in such a way that it can readily use the LinuxONE Pervasive Encryption capabilities for data at-rest encryption when scaling up your cloud infrastructure.

Hardware Security Module

To use Pervasive Encryption across your cloud infrastructure, the IBM LinuxONE platform uses dedicated Crypto-Express6S adapters for processing cryptographic workloads under the platform. Combined with the Central Processor Assist for Cryptographic Function (CPACF) co-processor, the IBM LinuxONE platform can provide high-speed encryption by removing the encryption overhead from the Integrated Facility for Linux (IFL) processors and routing them to its dedicated and specific hardware for such workloads.

In addition, the Crypto-Express6S adapter includes the HSM for securely storing and protecting encryption keys that is compliant with the NIST FIPS 140-2 Level 4 standard. These cryptographic co-processors are protected within a tamper-responsive environment that destroys encryption keys if it senses an attack.

IBM Secure Services Container

The IBM Secure Services Container for IBM Cloud Private is a solution that hosts container-based applications for hybrid and private cloud workloads on IBM LinuxONE. This secure computing environment for microservices-based applications can be deployed without code changes to security capabilities and provides the following benefits:

- ▶ Tamper protection during installation and start time to protect against malware attacks
- ▶ Restricted administrator access to help prevent the misuse of privileged user credentials for cloud and on-premises environments
- ▶ Automatic Pervasive Encryption of in-flight and at-rest data

The IBM Secure Services Container (SSC) technology is built on top of the workload isolation of the firmware-based Logical Partitions (LPARs). IBM LinuxONE LPARs adhere to the Common Criteria security evaluation EAL5+ and is unique to the platform. For example, IBM Cloud uses the advanced security mechanisms of IBM Services Container on its IBM Blockchain Platform and is extended for generic container-based applications through IBM Secure Services Container for IBM Cloud Private.

For more information see “IBM Secure Service Container for IBM Cloud Private” on page 229.

IBM Cloud Private security

IBM Cloud Private is a highly engineered system that is built with a security mindset. For more information about how to use a secure private cloud system across your infrastructure to meet the most demanding regulatory requirements, see 7.5.3, “IBM Cloud Private Security” on page 218.

IBM Cloud Private includes capabilities to monitor, secure, and operate microservices at scale. This feature helps customers to promptly monitor their infrastructure state in a centralized and more manageable way when compared to traditional on-premises infrastructures. IBM Cloud Private also uses Linux capabilities that are used by Docker containers, such as namespace isolation and control groups, for resource consumption limit and accountability.

Namespaces

Linux Namespaces are the most fundamental aspect behind containers. It is the fundamental security capability that allows for containers to be isolated one from another in multiple layers according to developers and business requirements.

Control Groups

Control Groups allocates and limits the resource consumption of processes in such a way that one workload does not affect the availability or performance of other workloads that are running in parallel.

With IBM Cloud Private, both capabilities can be used from within its management console to promptly isolate workloads while ensuring that misbehaving applications do not affect the whole cluster performance.

Moreover, the use of IBM Cloud Private with LinuxONE is strategic to organizations. The IBM LinuxONE Emperor II platform can scale up to 2 million containers in a single hardware footprint while providing all of its inherent security capabilities. All of this is achieved with unparalleled performance when compared to its competitors.

Containers and Orchestrator (Kubernetes) Security

One of the emerging problems with a container-based infrastructure concerns application image maintenance. Over time, failure to properly maintain an updated and private registry can introduce vulnerable containers within the cluster. Another concern relates to how the image was configured to make available its services and to which resources it has access.

As described in “Application Security” on page 219, the Vulnerability Advisor application provides mechanisms for customers to assess the compliance status of their images that are available within their registries. It also provides security reports from their currently running containers, which allows administrators to react quickly to possible security concerns that a container might introduce in the environment.

The private registry that is included with IBM Cloud Private also is frequently maintained and updated by IBM to mitigate all of the latest known vulnerability advisories. The recommendation is that customers always scale their workloads by using the latest images that are available within the IBM Cloud Private registry.

IBM Cloud Private also permits customers to introduce custom images to the private IBM Cloud Private registry. Whenever introducing custom images to an environment, the following precautions must be taken:

- ▶ Ensure that the imported image is patched against all latest known security advisories to date.
- ▶ Review with your security team whether the image satisfy your corporate policy requirements.
- ▶ Document and maintain security procedures for image maintenance and lifecycle.
- ▶ Avoid making available user credentials through APIs. Instead, use IBM Cloud Private Service IDs and API Keys, as described in “Service IDs and API Keys” on page 219.

Securing the IBM Cloud Private orchestrator is another important step to take to use the security features of your private cloud infrastructure. Always ensure that you use strict authentication and authorization mechanisms that conform with your organization’s security policy.

Also, ensure that you use privileged access revalidation mechanisms across your infrastructure to prevent unauthorized access to resources. IBM Cloud Private includes role-based access control (see 7.5.3, “IBM Cloud Private Security” on page 218) with which cluster administrators can grant access to resources based on the least privileged principle.

It is also important to split the network traffic into separated virtual networks based on the sensitivity of the traffic present within the workload. A common strategy is to prevent direct access to databases and middleware externally in such a way that only the application front end is made available for external access.

Another approach is to lock down workloads that are processing sensitive information and isolating those workloads from common workloads. With IBM Cloud Private and LinuxONE, customers use SELinux capabilities to harden their environments in such a way that only authorized operations are allowed, which makes it impossible for a potential attacker to gain access to privileged information.

Moreover, IBM Cloud Private allows for customers to use end-to-end encryption across their containers and microservices communications. Whenever possible, ensure that the cluster networks and deployed microservices use mutual authentication mechanisms in such a way that the network traffic can be established only when trusted by both parties.

Figure 7-13 shows the different layers for container-based applications and microservices that are deployed on IBM Cloud Private for IBM LinuxONE. Only the application front end is made available for external access, whereas the other supporting components are isolated from external access.

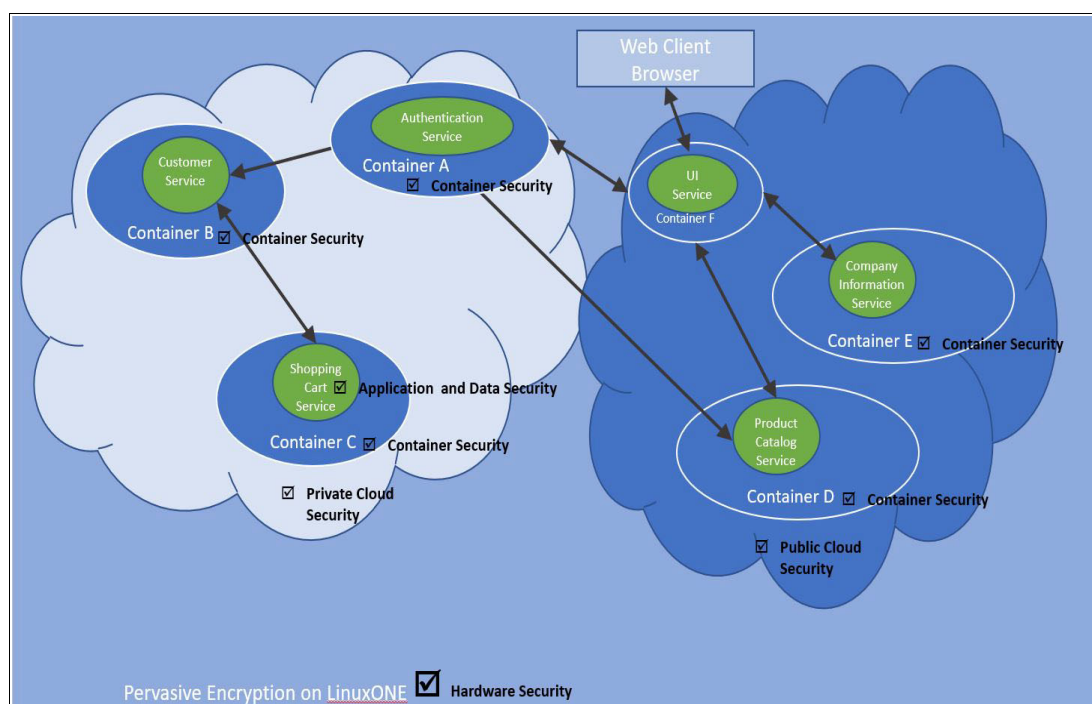


Figure 7-13 Security Levels for containerized applications on LinuxONE

Applications and microservices

Several features are provided to protect applications throughout their lifecycle. IBM Cloud Private combines secure engineering and secure deployment mechanisms to ensure that the private cloud is kept secure against internal and external threats.

IBM Cloud Private also provides multi-tenant application isolation and security for application runtimes and services. This capability is enhanced by the LPAR workload isolation that is included with the IBM LinuxONE platforms.

Also, high availability and business continuity management are brought together in a DevOps model. This combination helps customers to quickly and securely scale up their LinuxONE based workloads with ease.

By using IBM LinuxONE security features, such as Pervasive Encryption and IBM Secure Services Container for IBM Cloud Private, customers can use their private cloud security. The HSM helps customers securely store encryption keys that are tamper-proof against intrusion, which ensures that even physical access to encrypted resources are not possible.

All of these features are designed to protect applications and microservices and secure them from malicious use today and attacks in the future.

7.6.2 IBM Secure Service Container

IBM Secure Service Container provides the infrastructure to combine an operating system, middleware, and application components in a single software image. When the software image is deployed on a Secure Service Container partition, it can use certain security capabilities in the underlying infrastructure of LinuxONE servers, such as LPAR workload isolation, Pervasive Encryption of data at-rest and in-flight, and restricted root access to the environment.

By focusing on ease of management, ease of deployment, and security, the Secure Service Container is delivered in a virtual software appliance form factor. This factor can also isolate the running workload and deliver protections around the access of the environment.

The Secure Service Container is a general framework and serves as the service layer by integrating with IBM Cloud Private to manage the workloads that are deployed in IBM Cloud Private.

In the Secure Service Container, a specialized Docker runtime environment called `runq` is used to create a dedicated `qemu` virtual machine (VM) for each instantiated Docker image. This Docker runtime also provides for each of those `qemu` VMs a dedicated guest operating system kernel.

As shown in Figure 7-14 on page 229, all these components are packaged as a software appliance and can be deployed on a partition of an LinuxONE server in a single step.

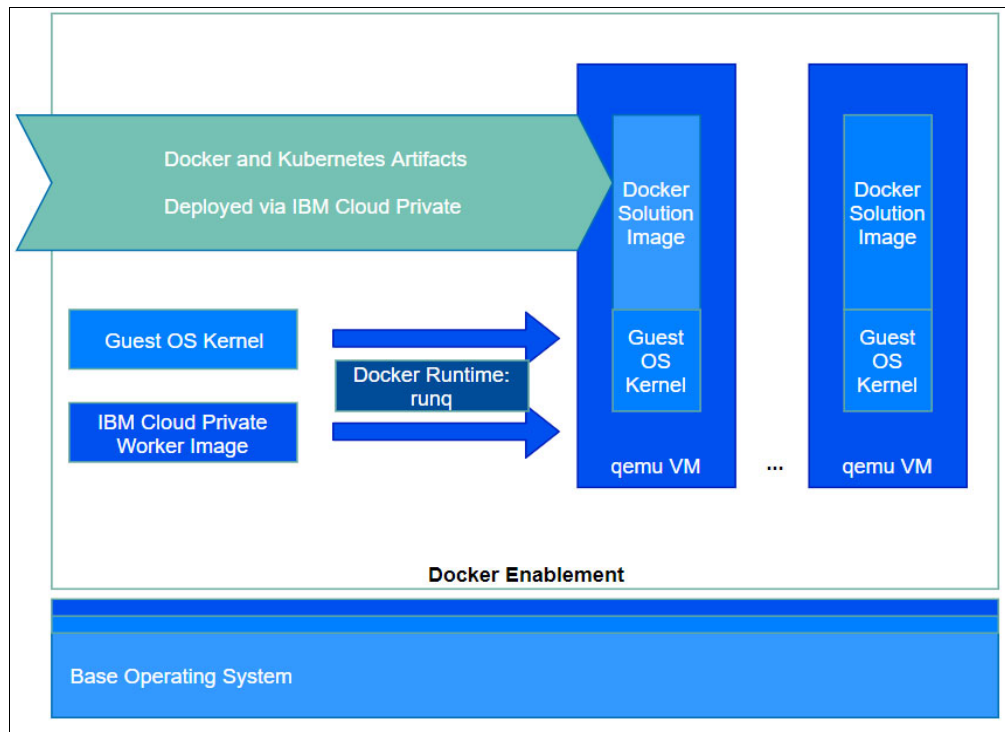


Figure 7-14 IBM Secure Service Container for IBM Cloud Private: Docker Enablement

IBM Secure Service Container for IBM Cloud Private

The IBM Secure Service Container for IBM Cloud Private is a software solution that is built on top of the IBM Secure Service Container framework that hosts container-based applications for hybrid and private cloud workloads on IBM LinuxONE. This secure computing environment for microservices-based applications can be deployed without requiring code changes to use the LinuxONE security capabilities and provides the following benefits:

- ▶ Tamper protection during installation time
- ▶ Restricted administrator access to help prevent the misuse of privileged user credentials
- ▶ Automatic encryption of data both in-flight and at-rest

IBM Secure Service Container for IBM Cloud Private provides an encrypted environment with peer-to-peer and peer-to-host isolation that protects container applications from access by using hardware and operating system administrator credentials, whether access is accidental or malicious, or internal or external to an organization.

Secure Service Container for IBM Cloud Private provides these protections while integrating with IBM Cloud Private. The IBM Cloud Private management stack delivers rapid innovation and application modernization, enterprise integration, and management and compliance to containerized applications.

Figure 7-15 on page 230 shows an overview of the full stack solution when running IBM Secure Service Container for IBM Cloud Private on top of LinuxONE. The Secure Service Container for IBM Cloud Private helps customers to deploy applications in the secure framework with ease on the LinuxONE platform.

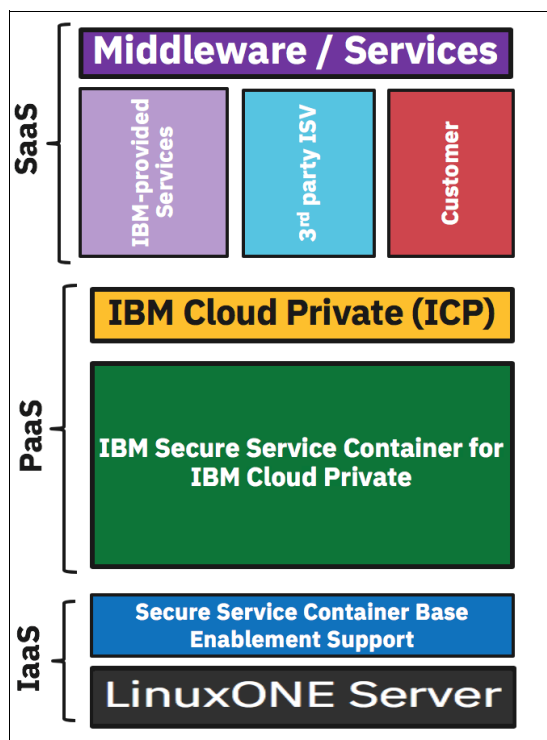


Figure 7-15 IBM Secure Service Container for IBM Cloud Private Full-stack Solution

7.6.3 Deploying IBM Cloud Private on LinuxONE

IBM Cloud Private is available in three editions to meet different business use cases. All editions include the base IBM Cloud Private product in addition to featured applications that are available through its catalog at no charge.

Before deploying IBM Cloud Private on your environment, it is important to understand how Kubernetes, Docker, and Helm work because these components form the core foundation of the IBM Cloud Private platform.

You use Kubernetes deployments to place instances of applications, which are built into Helm charts that reference Docker images. The Helm charts contain the details about your application, and the Docker images contain all the software packages that your applications need to run. For more information about these components, see the following resources:

- ▶ [Helm Documentation](#)
- ▶ [Docker - Get Started, Part 1: Orientation and setup](#)
- ▶ [Kubernetes Basics](#)

Community Edition

IBM Cloud Private Community Edition is available as a no-cost offering. The edition helps customers to get hands-on experience with platform and try it themselves within their own infrastructure as a proof of concept solution.

As with all other IBM Cloud Private editions, the Community Edition is built on top of a Kubernetes-based platform that includes core services and a content catalog of free services. Because this edition suited for proof of concept and hands-on with the IBM Cloud Private technology, it is intended for non-production workloads and is limited to one master node³.

Cloud Native Edition

Cloud Native Edition includes all of the capabilities of the Community Edition but allows for high available clusters to exist to support a production on-premises cloud infrastructure. Also, the services catalog is extended in such a way that key offerings, such as IBM Vulnerability Advisor, are also available starting with this edition. Cloud Native is suited for enterprise developers who want to modernize their applications manageability and operations.

Enterprise Edition

IBM Cloud Private Enterprise Edition further extends the previous editions and is suited for enterprises to modernize and optimize applications while scaling their data centers to securely work with cloud services. The services catalog is expanded to support more enterprise products, such as IBM WebSphere Application Server Network Deployment, IBM MQ Advanced, and IBM API Connect®.

IBM Cloud Private deployment considerations

IBM LinuxONE features unmatched scalability and security capabilities when compared to traditional x86 workloads. LinuxONE systems can meet all of the hardware and software requirements to deploy an on-premises cloud infrastructure within a single hardware footprint.

The IBM Cloud Private is built in such a way that it is flexible to meet various demands. In that sense, several architecture-specific decisions must be considered before proceeding with the product installation process. As described in 7.6.3, “Deploying IBM Cloud Private on LinuxONE” on page 230, it is necessary to understand the open source components that are used by IBM Cloud Private and how they interact with each other. This understanding helps to properly plan, size, and ensure that the cloud can meet the growth demands that are required to scale your infrastructure.

Along with the proper sizing and planning details that are specific to your IBM Cloud Private cluster, the following other aspects must also be considered:

- High Availability

IBM Cloud Private supports High Availability configurations to ensure that even the failure of critical cloud components do not interrupt IT and business operations. The IBM LinuxONE platform further uses this capability with z/VM Single System Image (SSI) and Live Guest Relocation (LGR) capabilities. LGR allows for Linux guests to be quickly relocated to other z/VM domains without any downtime, which ensures that the cloud can continue to run even during maintenance that otherwise require downtime.

- Networking

A proper network topology is required for most cloud workloads. If possible, a highly available external load balancer, such as an F5, can be used to spread the traffic among separate master or proxy node instances in the cluster.

Such approach allows for your IBM Cloud Private cluster to coexist across different subnets, which allows for continuous operation even when its primary network faces an outage. Other considerations that are related to networking involve the capabilities of a cluster to communicate with other IBM Cloud Private clusters, and other infrastructure components that are external to the IBM Cloud Private infrastructure.

Under IBM LinuxONE, z/VM can provide VLAN Aware VSWITCH capabilities to allow the cloud infrastructure to scale under different network segments within a single hardware footprint.

³ A master node provides management services and controls the worker nodes in a cluster. To support an HA solution, multiple master nodes must be deployed within a cluster.

► Storage

Planning a proper storage infrastructure is critical to attend the growth requirements in the digital era. In a world that is constantly producing structured and non-structured data, the backend storage facilities of a cloud must be readily enabled to support these growing demands.

IBM Cloud Private support several storage solutions, such as IBM Spectrum™ Scale and Ceph. Moreover, backup and restore mechanisms must be in place to ensure that the cloud can recover from disasters and ensure that business-critical data can quickly be restored if a loss occurs.

► Security

IBM Cloud Private provides several security mechanisms to secure your cloud infrastructure. In addition to role-based access control and Vulnerability Advisor (see 7.5.3, “IBM Cloud Private Security” on page 218), IBM Cloud Private provides mechanisms for data encryption in-transit and at-rest. Data in-transit encryption is achieved through the combination of TLS and IPSec mechanisms, whereas data at-rest can be employed at filesystem-level or block-device-level by using dm-crypt.

Also, it is important that the Linux guests that support a cloud infrastructure are hardened with mechanisms, such as SELinux and Two-Factor Authentication, to ensure that only authorized personnel can gain privileged access to such systems.

System integrity monitoring is an important part of several compliance and audit requirements, including Payment Card Industry Data Security Standard (PCI/DSS). IBM Cloud Private also provides the Mutant Advisor service, which tracks changes of a container in files and processes. The reports can then be reviewed by the security personnel to determine which changes are normal and can be ignored and identify possible threats that might need acting upon.

With IBM LinuxONE, the entire cloud infrastructure can be hardened and encrypted with minimal performance overhead thanks to the Central Processor Assist for Cryptographic Function (CPACF) co-processor. Also, customers that deploy Java workloads within the platform benefit from the new Galois Counter Mode (GCM) feature that is on the CPACF, which creates industry-leading Java performance by using TLS workloads.

► Other considerations

Other important factors include isolating and segmenting sensitive applications and how logging and auditing mechanisms are to be used across the cloud. IBM Cloud Private is a flexible platform that meets various business demands. Therefore, it can scale up small and use growth to meet even the most complex demands.

The IBM LinuxONE platform meets all of these demands in a single hardware footprint. The IBM LinuxONE Emperor II can support up to 32 TB of memory and up to 170 Integrated Facility for Linux (IFL) cores. It is also avoids or instantly recovers from failures to minimize business disruptions.

IBM LinuxONE security features, such as LPAR workload isolation, Pervasive Encryption, and IBM Secure Service Container, help customers to deploy and scale their microservices and applications within IBM Cloud Private in a secure and controlled way, which helps meet even the most regulatory compliance requirements on the market.

Note: For more information about installing IBM Cloud Private, see [IBM Knowledge Center](#).

7.6.4 IBM Cloud Private hands-on

Customers are encouraged to evaluate IBM Cloud Private in various ways. The Community Edition is available for customers to deploy the technology within their own data centers, which allows operations and development teams to build expertise and organization-specific procedures as they roll out their business transformation strategy in accordance with the corporate policies that are in place.

The IBM Cloud Private Community Edition image is freely available from Docker Hub. Example 7-1 shows how to retrieve the IBM Cloud Private-CE image from Docker public registry.

Example 7-1 IBM Cloud Private can be retrieved for installation with a docker single command

```
$ docker pull ibmcom/icp-inception:3.2.0
Trying to pull repository docker.io/ibmcom/icp-inception ...
sha256:f051d6343c5005db7dc80df78d8c8e98178143dbb8bdd15d5b8d03ef0c567ce7: Pulling
from docker.io/ibmcom/icp-inception
27402aa444ba: Pulling fs layer
50495ac9ea81: Pulling fs layer
4667be868145: Pulling fs layer
7a1bc0739ab9: Pulling fs layer
ff2203138bad: Pulling fs layer
d86db87240ee: Pulling fs layer
f1be305238c4: Pulling fs layer
7793d5534402: Pulling fs layer
ccae98b93c8d: Pulling fs layer
41d0212bb023: Pulling fs layer
b0e137749e57: Pulling fs layer
90343c106c2c: Pulling fs layer
27402aa444ba: Pull complete
50495ac9ea81: Pull complete
4667be868145: Pull complete
7a1bc0739ab9: Pull complete
ff2203138bad: Pull complete
d86db87240ee: Pull complete
f1be305238c4: Pull complete
7793d5534402: Pull complete
ccae98b93c8d: Extracting [=====> ] 15.59 MB/15.59 MB
41d0212bb023: Download complete
b0e137749e57: Download complete
90343c106c2c: Download complete
ccae98b93c8d: Pull complete
41d0212bb023: Pull complete
b0e137749e57: Pull complete
90343c106c2c: Pull complete
17c28ed6d871: Extracting [=====> ]
49.58 MB/139.9 MB
b406349d04fb: Download complete
```

After the Docker pull command is complete, query for the image status as shown in Example 7-2 on page 234. The tag and created columns shows the version of the product, and the date on which the image was built and made publicly available by IBM.

Example 7-2 Displays available images ready for use

\$ docker images				
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker.io/ibmcom/icp-inception	3.2.0	42dc2b4b7f2f	5 weeks ago	1.07 GB

Assuming that all previous planning considerations were taken as described in 7.6.3, “Deploying IBM Cloud Private on LinuxONE” on page 230, follow the installation process as described in the [IBM Cloud Private documentation](#) from that point forward.

Figure 7-16 shows the IBM Cloud Private-CE welcome page after the installation process is completed. Customers are encouraged to use the product within their own IBM LinuxONE systems.

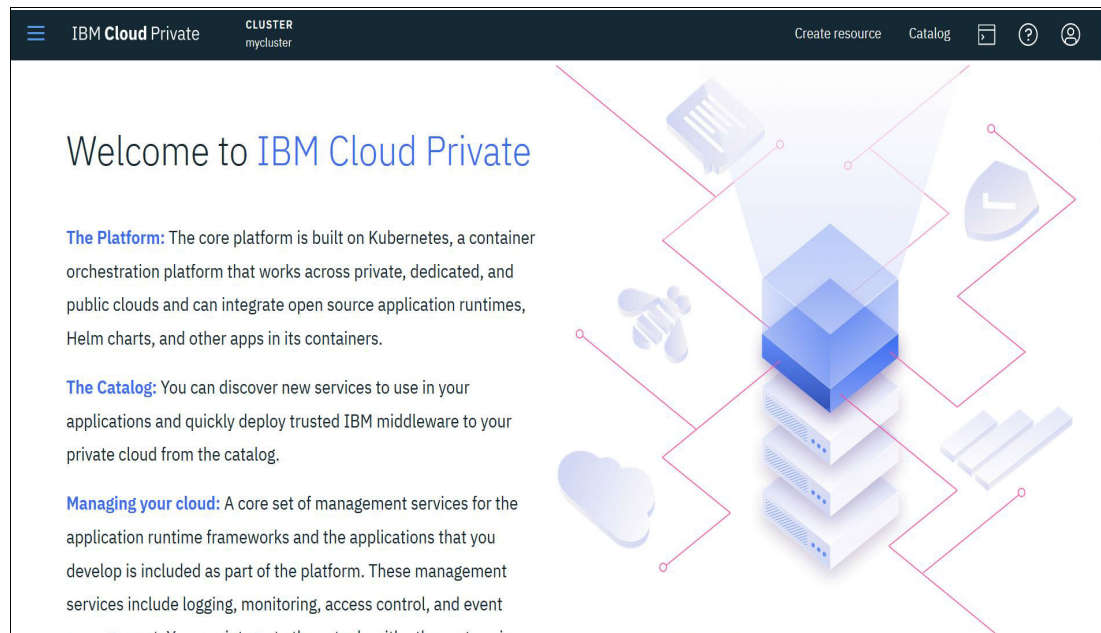


Figure 7-16 IBM Cloud Private welcome page

IBM Cloud Private guided demo

IBM acknowledges the fact that some organizations might not dispose of a readily available infrastructure to support all the necessary components of an enterprise private cloud. Some organizations might also have strict regulations that do not allow them to roll out IBM Cloud Private in a timely manner. Also, IBM also permits business leaders to have a readily available hands-on experience with the IBM Cloud Private platform without the need to use their own data center resources.

Customers are encouraged to use the [IBM Cloud Private guided demonstration](#) to quickly deploy a cloud-native application in Kubernetes running on top of the IBM Cloud Private platform. The guided tour that is provided by IBM Garage shows how quickly and securely a user can scale out their infrastructure within minutes.

7.6.5 Deploying a Node.js service on top of ICP and LinuxONE

This section describes how quick and simple it is to scale an infrastructure running on top of LinuxONE with IBM Cloud Private. In this section, we describe how to spin up a sample Node.js container from within the IBM Cloud Private GUI.

After logging-in to your IBM Cloud Private cluster, click in the upper left corner, as shown in Figure 7-17. The main menu of IBM Cloud Private offers various options to manage your infrastructure.

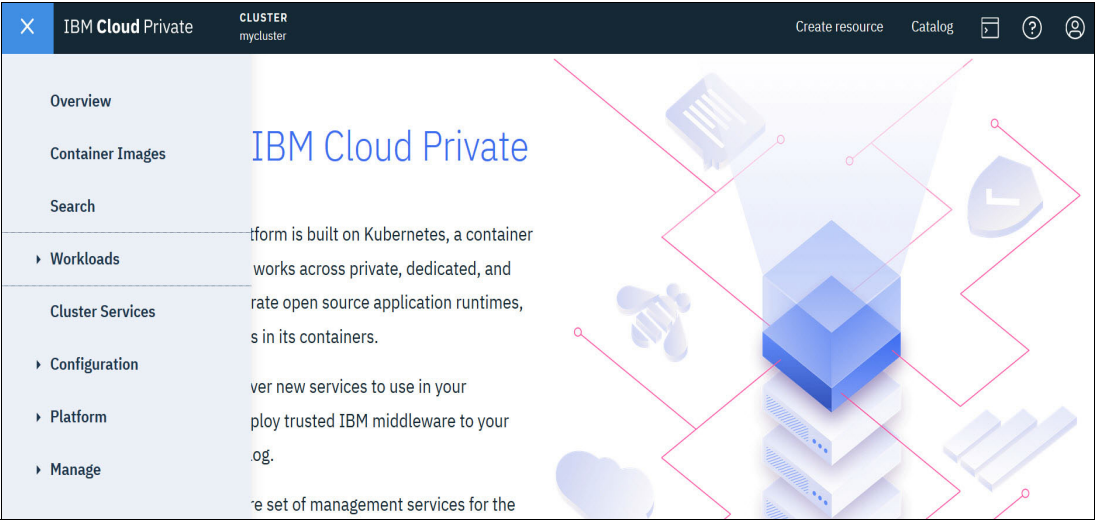


Figure 7-17 IBM Cloud Private Menu

Figure 7-18 shows an overview of the general status of your cluster. You can get to this page by selecting the **Overview** option from the IBM Cloud Private menu. The Overview page displays in an easy and intuitive way the overall status of the resources that are allocated to your cluster. Administrators use this option to quickly identify and predict problems.

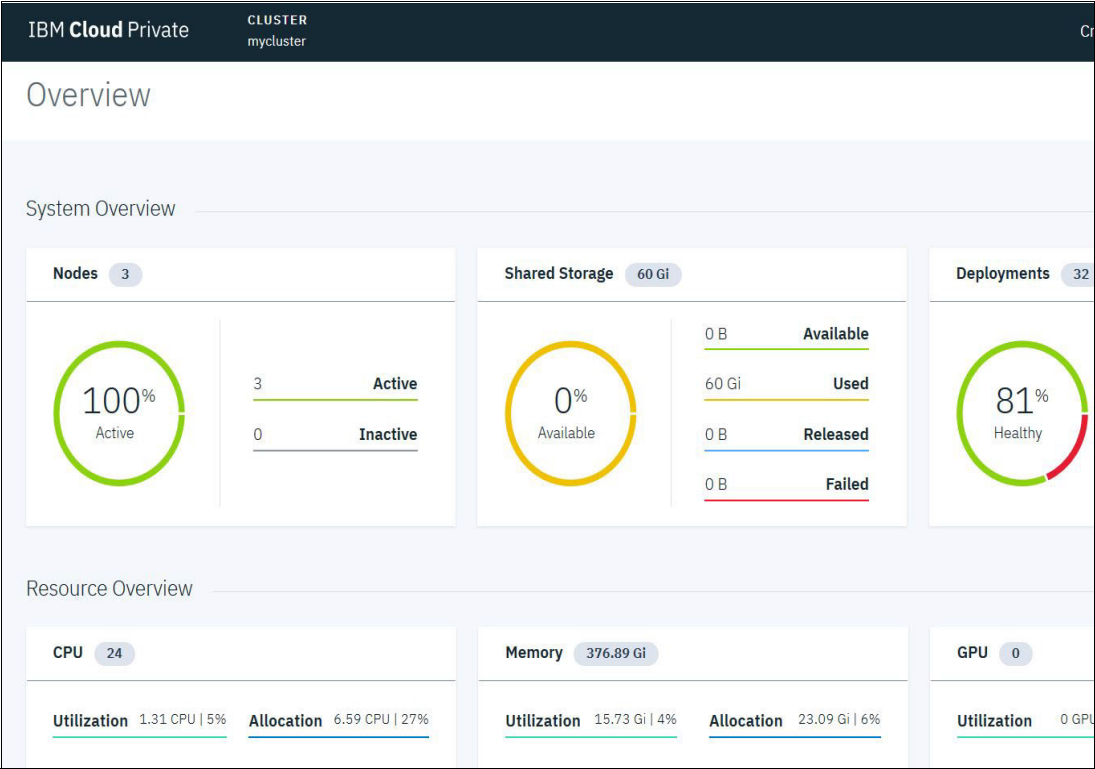


Figure 7-18 Cluster Overview window

Figure 7-19 shows the IBM Cloud Private catalog, which is accessible by clicking in the upper right side of the IBM Cloud Private GUI. The catalog includes services that are categorized in various areas, such as AI, IBM Blockchain, DevOps, IoT, and Security.

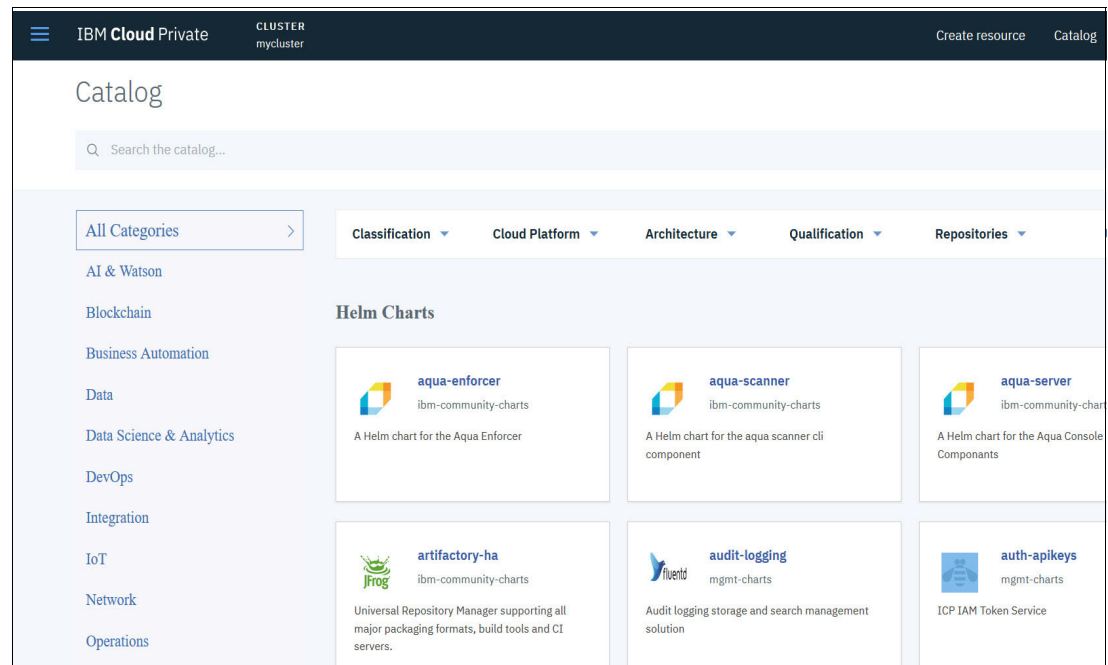


Figure 7-19 IBM Cloud Private catalog

To create the Node.js service, search for nodejs from within the catalog, as shown in Figure 7-20.

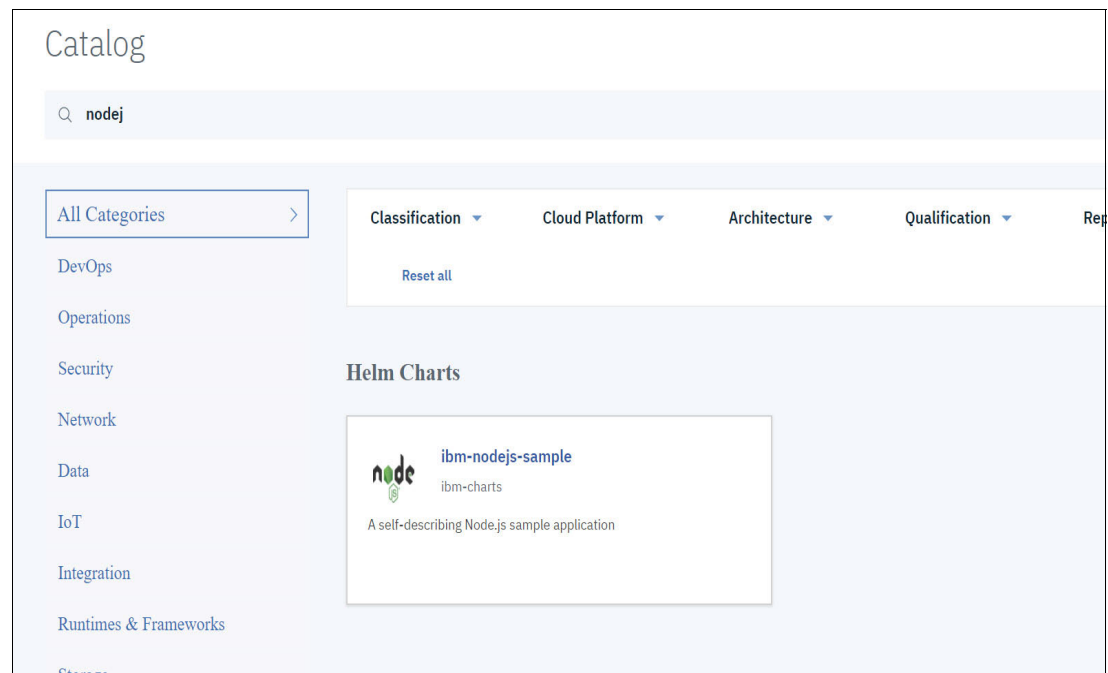


Figure 7-20 Search present in the IBM Cloud Private catalog

By clicking the wanted service to be deployed, the service's Overview page opens, as shown in Figure 7-21. The Overview page displays important information about the selected service and depending on the type of service that is deployed, the limitations and accepted configurations parameters. Click **Configure** in the lower right corner to proceed with the service configuration process.

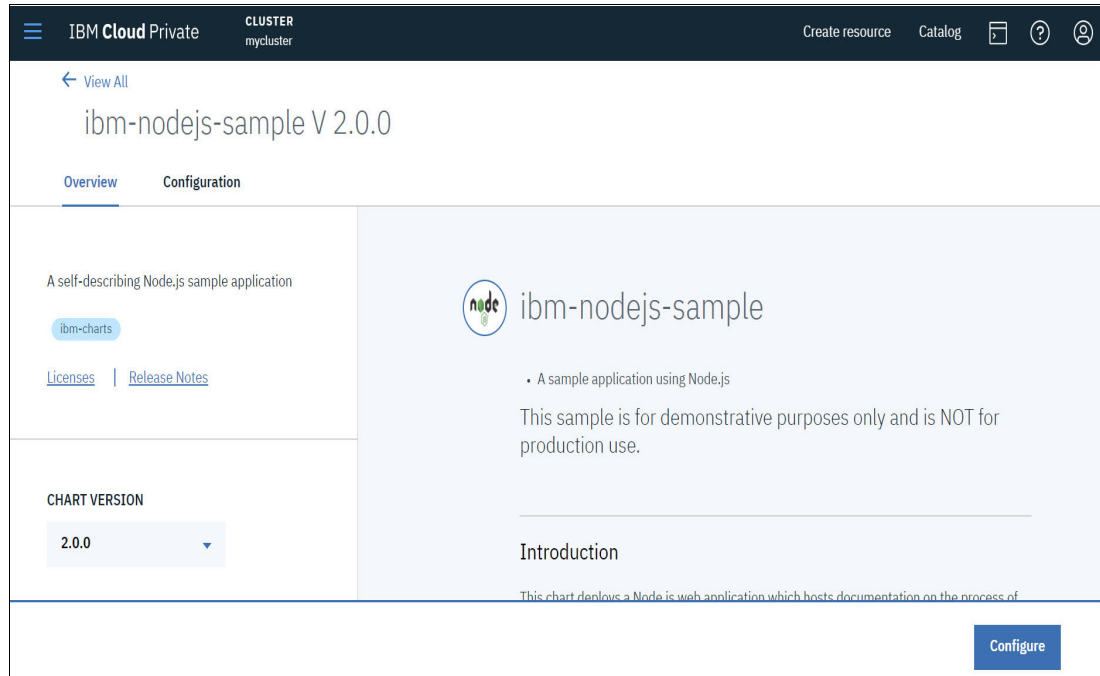


Figure 7-21 Node.js demo overview window

Enter the Helm release name and select the target namespace and target cluster to which you want the application deployed, as shown in Figure 7-22.

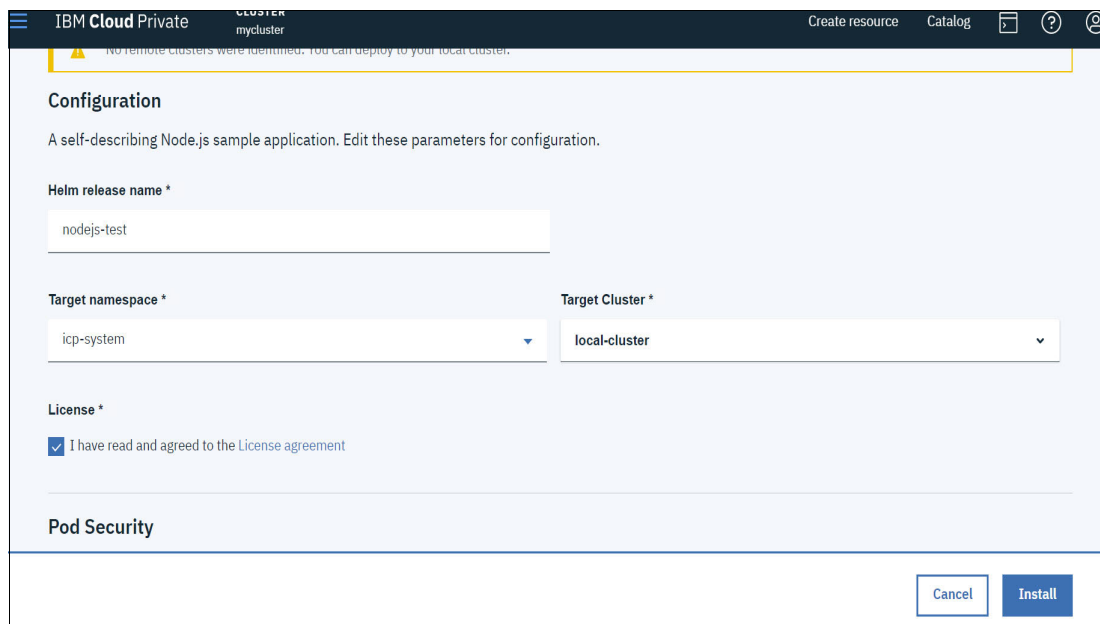


Figure 7-22 Sample Node.js application configuration window

Several other parameters can be adjusted before you proceed with the installation, such as Pod security, scaling options, and resource limits. These options are available by scrolling down the window.

When you are done with your customizations, click **Install** in the lower right side of the window. A pop-up window opens that indicates your deployment started, as shown in Figure 7-23.

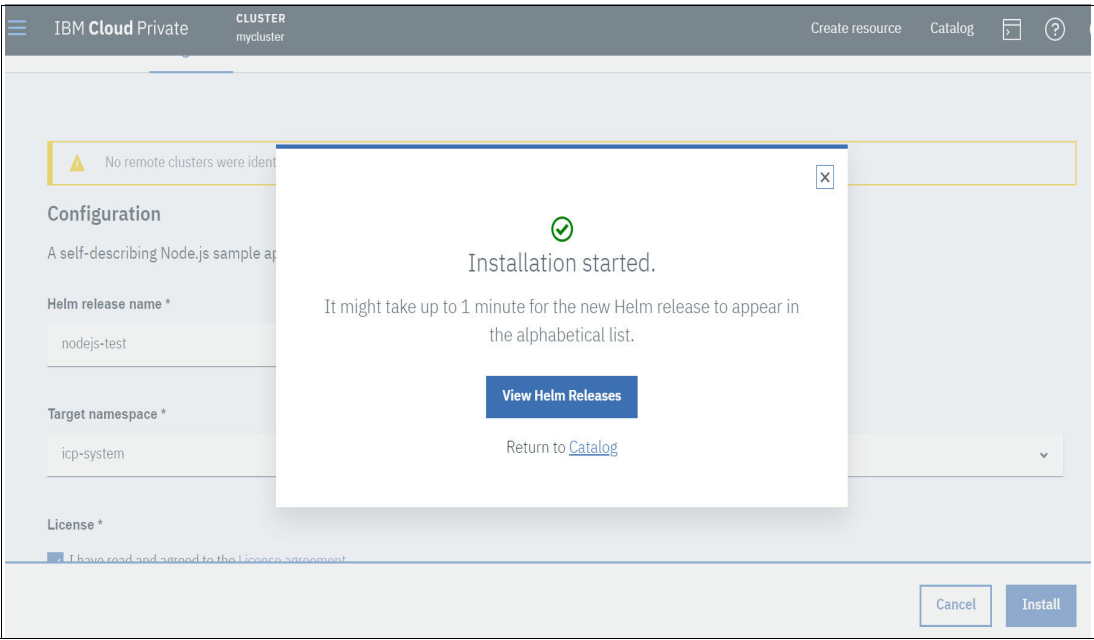


Figure 7-23 A pop-up appears indicating that the service installation began

By clicking **View Helm Releases**, you can check the status of your current deployment, as shown in Figure 7-24.

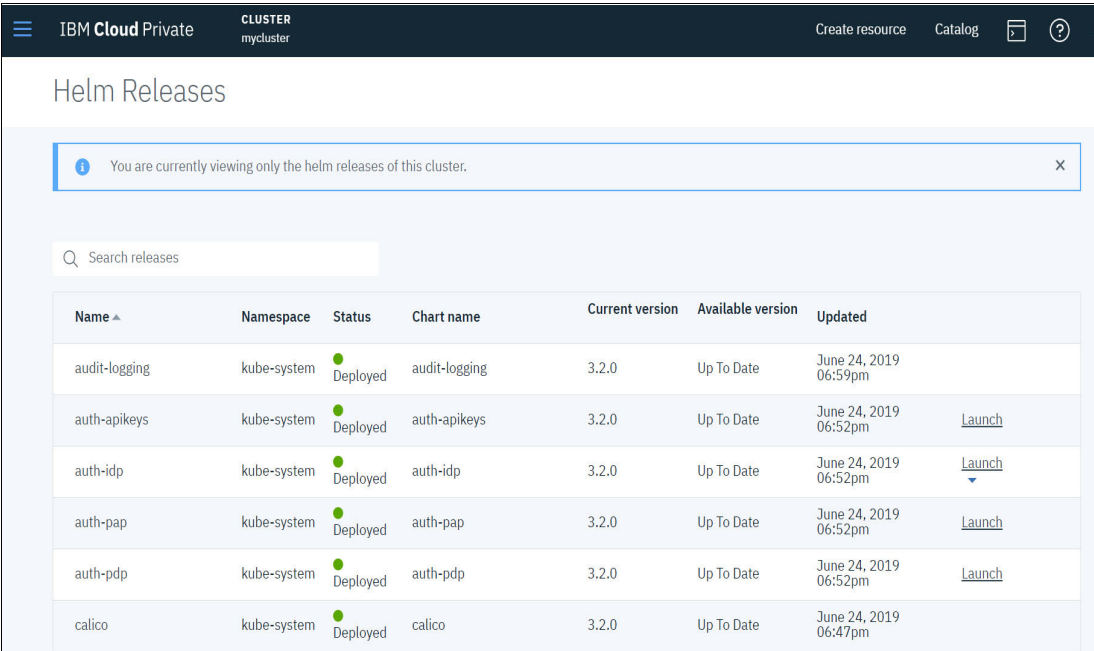


Figure 7-24 Helm Releases section on ICP displays the status of running containers

From there, you can see the status of the Node.js service that was just deployed. Figure 7-25 shows that the service was successfully started and is ready to be used.

monitoring	kube-system	● Deployed	ibm-icpmonitoring	1.5.0	Up To Date	June 24, 2019 06:58pm	Launch
multicluster-hub	kube-system	● Deployed	ibm-mcm-prod	3.2.0	Up To Date	June 24, 2019 06:59pm	Launch
nginx-ingress	kube-system	● Deployed	icp-nginx-ingress	3.2.0	Up To Date	June 24, 2019 06:48pm	Launch
nodejs-test	icp-system	● Deployed	ibm- nodejs -sample	2.0.0	Up To Date	July 4, 2019 10:36am	Launch
nodejs test	icp-system	● Deployed	ibm- nodejs -sample	2.0.0	Up To Date	June 24, 2019 07:38pm	Launch

Figure 7-25 Sample Node.js application has been successfully been deployed

Finally, click **Launch** to access the deployed application, as shown in Figure 7-26.

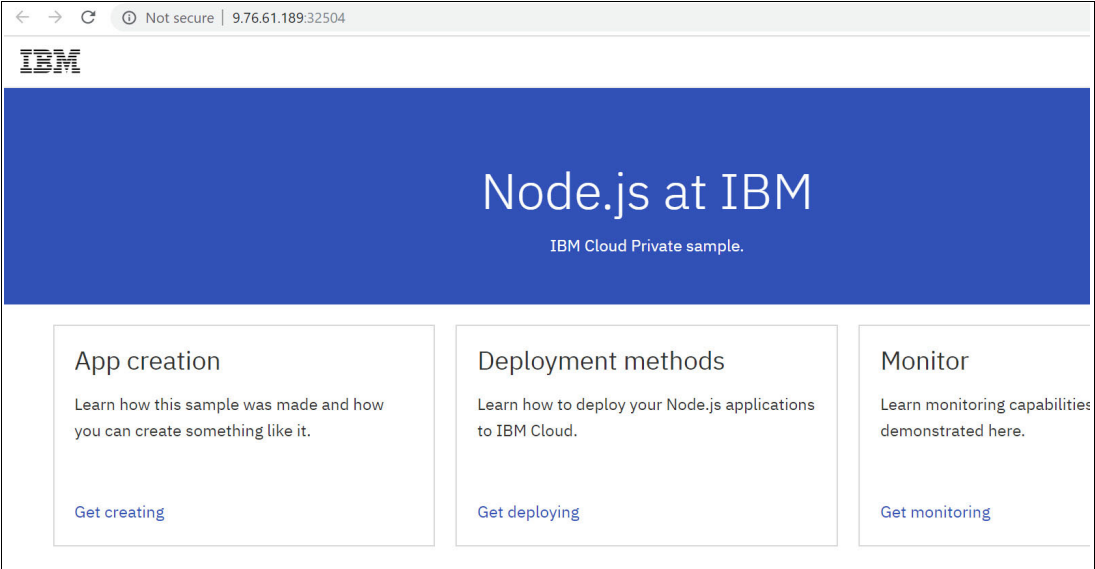


Figure 7-26 Node.js sample application is now up and running

7.7 IBM Cloud Automation Manager

IBM Cloud Automation Manager is a multi-cloud management solution in IBM Cloud Private for deploying a cloud infrastructure with an optimized user experience.

Although IBM Cloud Private can orchestrate a Container as a Service (CaaS) infrastructure on top of nodes, IBM Cloud Automation Manager uses open source IBM Terraform® to manage and deliver a full-stack cloud infrastructure that is presented as code.

Cloud infrastructure that is delivered as code is reusable and can be placed under version control, shared across distributed teams, and used to easily replicate environments. By using IBM Cloud Automation Manager, you can easily scale your LinuxONE infrastructure and automate several steps during its process.

With IBM Cloud Automation Manager, you can provision cloud infrastructure and accelerate application delivery into various cloud environments, such as IBM Cloud and OpenStack, with a single user experience. Because IBM Cloud Automation Manager is compatible with many cloud solutions, customers can also integrate their infrastructure into a single, centralized, and standardized solution for cloud infrastructure delivery.

You can spend more time building applications and less time building environments when cloud infrastructure is delivered by using automation. By providing a list of useful templates with the product, customers can get started faster by using the IBM Cloud Automation Manager library.

7.7.1 Terraform

Terraform is the de facto standard for declaring a wanted state for various clouds. By defining infrastructure as code, it reduces costs, increases speed to market, and improves security. IBM strategy in using Terraform as its backend solution for IBM Cloud Automation Manager allows organizations to use DevOps practices while complying with the IT security policies in place.

Although Terraform was initially targeted at Infrastructure as a Service (IaaS) deployments, it can also support other cloud offerings, such as PaaS and SaaS services. By defining infrastructure as *resources*, such as network switches, containers, and virtual machines, Terraform can create, manage, and update these components to speed up IT infrastructure changes and consolidate an organization's infrastructure.

The use of Terraform with Cloud Automation Manager has the following benefits:

- ▶ **Environment consistency:** Because the infrastructure is defined as code, the same configuration for staging, development, and production environments is achieved.
- ▶ **Prone to human errors:** Several steps must be taken care before a service is placed into production. By using automation, it is possible to ensure that all required processes are followed during the deployment of a stack.
- ▶ **On-demand provisioning:** By spinning up environments whenever needed, on-demand provisioning can reduce infrastructure resources that do not require to be available always, such as development environments.

7.7.2 IBM Cloud Automation Manager on IBM Cloud Private

IBM Cloud Manager uses available APIs to communicate with the underlying infrastructure and external services. By using this approach, customers also can integrate their business processes throughout its deployment stacks. For example, it is possible to ensure that every deployment is properly recorded within the company's asset central inventory. Another example includes automatic DNS registration and IP assignments for the newly provisioned assets.

It is also possible to create decision points throughout the provisioning of services, which makes the stack's deployment flexible to meet various situations. For example, if no IP assignments are left for a subnet, the provisioning can automatically select the next available subnet. If a critical error occurs, it can automatically notify the operational teams to quickly remediate the situation.

By using Cloud Automation Manager on IBM Cloud Private to create and edit templates and services that implement common business patterns, a software-defined infrastructure can be created that can be provisioned whenever the need arises. After the services are deployed, the instances can be managed, accessed, maintained, and shared across multiple operational teams directly from the Cloud Automation Manager user interface in a simple, quick, and intuitive way.

IBM Cloud Automation Manager with LinuxONE

IBM LinuxONE is a platform for high-volume transaction processing and large-scale consolidation. Because of its highly engineered virtualization architecture, LinuxONE simplifies the management of its underlying infrastructure and can provide an abstraction of several physical and logical components.

The benefits of the use of IBM Cloud Automation Manager for IBM Cloud Private on top of LinuxONE are many. The platform can quickly redistribute system resources when a spike occurs and scale up, scale out, or both as necessary, to prevent system crashes and slow response times.

The LinuxONE hardware is built for redundancy and high availability in such a way that the failure of a component does not affect the overall system's operations. Also, by using the Linux operating system and open source tools, the platform is the de facto standard for housing cloud-based workloads with industry-leading security and performance.

IBM Cloud Automation Manager can provision guests on top of LinuxONE in the following ways:

- ▶ Linux on LPAR mode (by using OpenStack DPM drivers)
- ▶ Linux guests under z/VM (by using z/VM Cloud Manager Appliance)
- ▶ Linux guests under KVM (by using Open Stack)

Because the LinuxONE platform can abstract and simplify several components of an infrastructure, deployment of the IBM Cloud Automation Manager and IBM Cloud Private products within the platform are strategic to an organization's success. As described in "IBM Cloud Private deployment considerations" on page 231, the IBM LinuxONE platform provides the necessary components to build, secure, maintain, and manage a private cloud within its single hardware footprint.

Figure 7-27 shows the components of IBM Cloud Automation Manager on IBM Cloud Private. It provides automated provisioning of infrastructure and applications with workflow orchestration and integrates with various cloud environments.

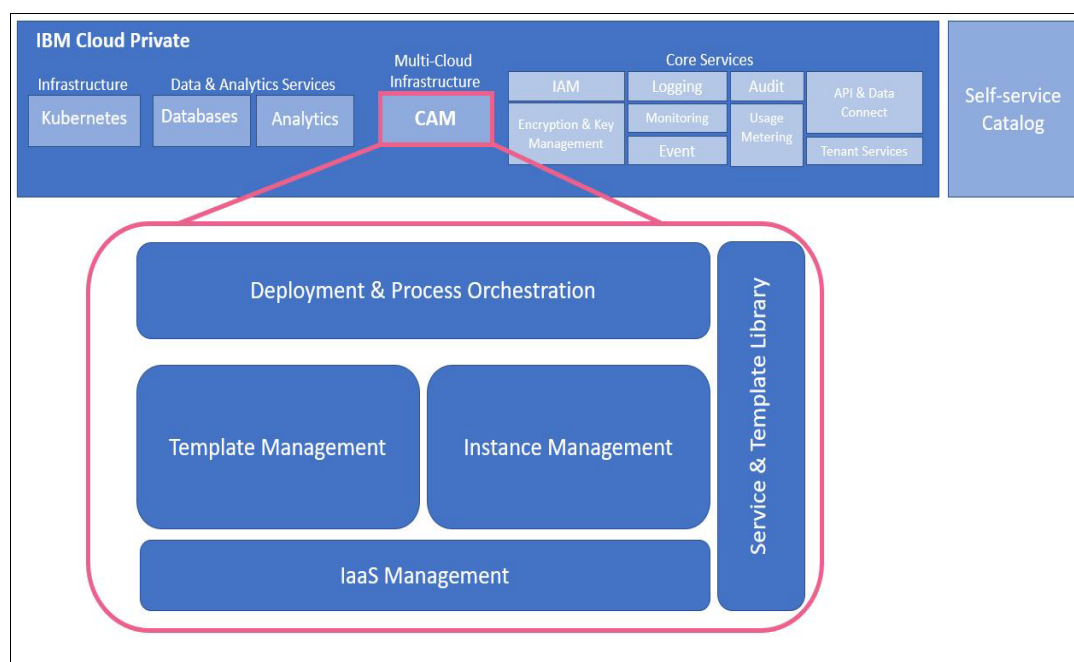


Figure 7-27 IBM Cloud Automation Manager on IBM Cloud Private

7.7.3 Security in IBM Cloud Automation Manager

Because IBM Cloud Automation Manager is installed within an IBM Cloud Private infrastructure, it automatically inherits all of the security benefits, as described in 7.5.3, “IBM Cloud Private Security” on page 218. In addition, because Cloud Automation Manager plays a key role for automating infrastructure deployments, many security aspects require special attention.

As it happens with any enterprise critical application, access to IBM Cloud Automation Manager must be locked down so that only administrators and developers with a valid business need can modify the deployment pipelines. Guaranteeing that the services and templates that are created are compliant with the governing security policy is also important to ensure that the deployed infrastructure is resistant against internal and external threats. In that sense, make it a common practice that your security team frequently audits and reviews the code created to provision your on-premises infrastructure.

Whenever IBM Cloud Automation Manager is used to deploy a new component to a production infrastructure, ensure that the following requirements are met:

- ▶ The deployed resource is patched and up to date against all latest known security advisories. Also, consider the use of regularly scheduled patching windows for your services by using tools, such as IBM BigFix®.
- ▶ A health check process occurs before service hand-over to users.
- ▶ Automatically change all default passwords that are set by the vendor. Use key-based authentication mechanisms instead of hardcoded passwords.
- ▶ Avoid storing passwords in clear text in scripts, recipes, or any other process as required by the organization.

It is a good practice to segment the corporate network in such a way that the development and production infrastructures are isolated one from another. Similarly, place workloads processing sensitive data on their own restricted networks. Such systems must be hardened to lock down any intrusion attempt. Some examples of security hardening include SELinux and Pluggable Authentication Modules for the Linux operating system, and IBM Guardium® for databases and big data workloads.

Adopting a centralized logging infrastructure is also critical to quickly identify possible threats and extraneous events. By using solutions, such as IBM QRadar®, logs can be captured from your cloud and its supporting components. Similarly, consider the use of tools, such as the Linux File Alteration Monitor (FAM) or Mutation Advisor from IBM Cloud Private, to quickly identify changes to critical operating system files that can pose your business at risk.

Also, consider safe-guarding API keys that permit access to critical components of your organization. Implement security mechanisms to frequently refresh such credentials to ensure that only authorized parties can use them.

IBM Cloud Automation Manager is a powerful multi-cloud solution that can deliver infrastructure as code across many different cloud providers. It uses DevOps practices and integrates multiple processes within an organization into a much simplified and agile way.

Along with IBM Cloud Private, both solutions together are the basis of an enterprise grade cloud platform. As you roll out the cloud transformation across your organization and develop processes and integration across your infrastructure, it is important to prioritize security by employing the techniques and practices that are described on this chapter.

Also, the union of IBM Cloud Private with the security capabilities that are provided by the IBM LinuxONE platform ensure that customers can scale their environments in an agile way and rely on a set of industry-leading security features.



IBM z/VM and enterprise security

This chapter provides information about IBM Security zSecure™ Manager for IBM Resource Access Control Facility (RACF) z/VM (zSecure). It also describes LDAP on z/VM and Linux on IBM LinuxONE security.

This chapter also introduces zSecure and provides a brief overview with some benefits of using this feature. zSecure supports ease-of-administration of RACF and helps you meet audit demands.

Some aspects of using LDAP on z/VM also are described, including how it can form part of enterprise security management.

The chapter also describes Linux IBM LinuxONE security from the perspective of the following items:

- ▶ Access control
- ▶ Audit issues
- ▶ Cryptographic functions available
- ▶ User Management.

This chapter includes the following topics:

- ▶ 8.1, “z/Secure” on page 246
- ▶ 8.2, “Lightweight Directory Access Protocol” on page 246
- ▶ 8.3, “Linux on IBM LinuxONE security” on page 249

8.1 z/Secure

zSecure is an optional product that can help simplify administrative tasks.

It supports administrators in enabling more efficient and effective ways of setting up profiles and group structures in your RACF databases. Recurring administrative tasks can be automated by using zSecure. It helps minimize complexity and improve quality of service. Reports can be taken regularly or even on an automated basis, so changes to the RACF database can be visualized by comparing the reports.

Audit functions are included in the product, which provides material to auditors and meets auditors demands for reports.

zSecure supports the following issues:

- ▶ Adding or deleting user IDs and groups
- ▶ Granting access to user IDs and groups
- ▶ Setting and resetting user IDs and passwords
- ▶ Running daily, weekly, and monthly reports

The audit functions of zSecure help identify potential security concerns and prioritize them by ranking the concerns that are identified. Inconsistencies in the security definitions or missing definitions can be addressed quickly. Vulnerabilities can be detected before they raise a serious security issue.

With the CARLa Auditing and Reporting Language (CARLa), you can create your own reports. These reports can be run under IBM Interactive System Productivity Facility (ISPF) or in batch by using data from any RACF database, or live or extracted RACF System Management Facility (SMF) data.

For more information about zSecure, see the product documentation at [IBM Knowledge Center](#).

8.2 Lightweight Directory Access Protocol

Lightweight Directory Access Protocol (LDAP) is a commonly used system for storing directory-style information, such as (user records) in a centralized database. Most directory and authentication systems use LDAP in some way to make directory information available for services such as network-based authentication.

Note: Microsoft Active Directory is based on Kerberos, a system for cryptographically secured access control in a distributed network, and LDAP. Other identity management systems, such as the Open Source *FreeIPA*, are also based on LDAP and Kerberos.

8.2.1 LDAP on z/VM

z/VM TCP/IP contains an LDAP server. This LDAP server supports secure access by using TLS, and has the following storage back ends that can present directory information:

SDBM (RACF DB)	SDBM provides access to the RACF database by using a custom LDAP schema. A record of activities that are done on this back end is automatically maintained in the RACF audit stream.
LDBM (Byte FS)	The LDBM back end stores data and schema in files in the Byte File System. In addition, it supports the use of RACF as the password store. The schema is easily extensible.
GDBM (logging)	GDBM provides an audit and log capability for LDBM. Normal LDAP search operations can be done to review changes that are made to the directory data that is stored in LDBM.
CDBM (configuration)	This back end stores the configuration of the LDAP server. The configuration can be updated by using LDAP modify operations, and the back end can be used in a clustering configuration to replicate configuration changes to other cluster members.

For more information about setting up the z/VM LDAP server, see Chapter 3, “Configuring and using the z/VM LDAP server”, in *Security for Linux on System z*, SG24-7728.

z/VM LDAP and authentication in your cloud

The z/VM LDAP server is lightweight and highly accessible in a z/VM Single System Image (SSI) environment. The ability to use the RACF password store (directly with SDBM or through native authentication with LDBM) makes it a secure option for password management. Passwords are stored in a highly secure way, and a high level of auditing is available by using the z/VM LDAP server.

In 3.8, “Using an OpenLDAP server with the z/VM LDAP server” in *Security for Linux on System z*, SG24-7728, the authors describe many ways that you can use the z/VM LDAP server to augment the security of an authentication environment that is centered on LDAP. Those techniques are summarized here.

Using z/VM LDAP with RACF as a password vault for other LDAP servers

The OpenLDAP server supports features that allow a z/VM LDAP server with RACF (directly with SDBM or through native authentication in LDBM) to be used as a separate and secure password store. Any LDAP server that supports the rewriting of LDAP URIs can use this technique.

Using RACF as a password store has the following benefits:

- ▶ Passwords are no longer stored in the LDAP database. A compromise of the database that is stored on the Linux system does not yield password data that can be used by an attacker to further compromise the environment.
- ▶ Password management issues such as password reuse and expiry are handled by RACF, without any support required in the other LDAP database.
- ▶ Auditing of authentication requests can be done by using the RACF audit stream.
- ▶ RACF provides a high degree of resilience and recoverability for the password store.

Using z/VM LDAP directly for Linux authentication

The LDBM back end of the z/VM LDAP server can support common applications, including Linux authentication. The schema that is supplied by IBM with the z/VM LDAP server supports Linux authentication directly, but the schema can easily be extended by using schema files from other LDAP servers, such as OpenLDAP.

If LDBM is used with native authentication, then the benefits above << above what?>> regarding separation of LDAP and password data also apply.

8.2.2 Integration of z/VM LDAP into an enterprise directory

Every sufficiently large organization experiences the issue of having to manage the distribution of directory information between departments, or between directory servers of different technologies. This is especially true with z Systems installations where RACF on z/VM or z/OS might be used.

The thought that an enterprise directory must be stored on a single monolithic server and managed by a single application is not true. A good directory structure supports the ability for portions of the directory tree to be held in different servers according to geographic, organizational, or technical reasons. Likewise, a directory based on LDAP can be distributed across any number of LDAP implementations. This concept is a fundamental part of LDAP: The *directory information tree* (DIT) refers to the entire structure of an enterprise directory, encompassing parts of the directory that might be widely distributed across the enterprise.

It is not likely that z/VM LDAP is used as the core directory store for an enterprise. However, for applications on Linux guests under z/VM, it makes an excellent choice due to its lightweight, proximity to the Linux systems, and password security.

Resource usage

The z/VM LDAP server is defined to the CP directory with a main storage size of 128 MB, and the disk space that is used by the LDAP server is already defined as part of the z/VM installation. Typical Linux guest configurations start at 1 GB main storage and at least 20 GB of disk space. Using z/VM LDAP for this function is much more lightweight than using a Linux virtual machine (VM).

Locality of reference

In computer science, *locality of reference* refers to how far through its cache and memory structure a processor must descend to reach an item of data. In directory management, you can use the term to describe how far away a directory is from the application referring to it.

Performance

If an application must make many references to directory information, keeping that directory as close as possible to the application might have positive performance benefits.

A directory tree can be built so that the parts of the directory that are relevant to applications that are hosted on systems under z/VM is stored on the z/VM LDAP server. In addition, through referral records and other methods, systems across the enterprise can access all corporate directory data regardless of where it physically is. Systems within the z/VM environment still can access other parts of the directory, and systems in other parts of the business can access the z/VM-hosted parts of the directory, as required.

Availability

Locality of reference also has an availability aspect. Bringing the portion of the DIT that is essential to application operation on to a server image that is physically and logically close to the application reduces the number of failures that might affect the application's operation. z/VM LDAP in an SSI environment can be made highly available through the following important features:

- ▶ Automatic sharing of the RACF database in SSI (for SDBM, and native authentication behind LDBM)
- ▶ z/VM LDAP replication (for an LDBM database)

Using z/VM LDAP as the directory store for application-critical directory data can improve the reliability and availability of the applications. The applications running on Linux guests in that environment can be configured to access z/VM LDAP directly over a vSwitch or guest LAN, eliminating any dependency on external network components for that access.

Directory integration

If it is not feasible to make a clean break between portions of the DIT, *directory integration* is required. Directory integration is the practice of making updates across various directories.

IBM Security Directory Integrator

The integration of directory information between RACF and other directory systems can be achieved by using technologies such as IBM Security Directory Integrator. Formerly known as IBM Tivoli® Directory Integrator, IBM Security Directory Integrator allows for the controlled propagation of directory information, including changes to existing records, between different directory systems.

IBM Security Directory Integrator provides a wide range of connectors for different types of directories. It uses a workflow-based methodology that is described as The AssemblyLine to manage the data flowing between information sources and targets, and how that data is transformed along the way if needed.

Note: For more information about IBM Security Directory Integrator, see [IBM Knowledge Center](#).

8.3 Linux on IBM LinuxONE security

Running the enterprise cloud with Linux on IBM LinuxONE servers provides some advantage over a physically distributed server farm when unique technologies within the platform are used to harden the overall security. In this section, the security issues that must be considered when implementing a Linux environment on your enterprise cloud are described.

8.3.1 Authentication

Authentication is the process of determining whether someone really is who they claim to be. The users attempting to access a system or a resource must first give sufficient proof of their identity.

A server authentication is done by using two of the following three categories, or factors, for providing identity:

- ▶ Something that you know: A password or PIN
- ▶ Something that you have: A token, a user ID, a badge, or a certificate
- ▶ Something that you are: Biometrics characteristics

The traditional way to authenticate on a Linux server is having a user ID on the system and knowing a password for it. Implementation of more than two of the factors is available for a Linux system, making it possible to request a user ID, a password, and a PIN that is generated by some electronic device, such as a token or an application on a cell phone, when authenticating. Use of more factors for authentication brings a high level of security to what is being accessed.

The Pluggable Authentication Modules (PAMs) can be used to reinforce compliance with the organization information security policy by increasing the number of factors that are used to authenticate and allowing access only to users meeting the specific characteristics that are defined with PAM. Applications that are enabled to use PAM can be plugged into new technologies without the need to modify the existing applications. This flexibility provides administrators with the following advantages:

- ▶ Use any available authentication service for an application.
- ▶ Use multiple authentication mechanisms for a service.
- ▶ Add authentication service modules without needing to modify the application.
- ▶ Use a single password for authentication on multiple modules.

Another factor that improves the security level is the use of PKI, such as an SSH key pair. The users must have a public and private key pair that ensures the user's ID. Although it is possible to use an SSH key pair without setting a password to it, a password should be set to the key pair. This prevents an attacker who has access to the private key but does not know the password for the user ID from being authenticated at the server.

8.3.2 Access control

Defining each job role in Linux is complicated because everything converges to the root user ID. However, doing so is a preferred practice that defines the access and provides strong control over who can access a superuser account.

The discretionary access control (DAC) model, which is standard Linux security, does not provide protection from broken software or malware running as a normal user or root. Users can grant risky levels of access to files they own.

Use of mandatory access control (MAC) provides full control over all interactions of the software. Administratively defined policy closely controls users and process interactions within the system, and can protect the system from broken software or malware running as any user.

Security-Enhanced Linux (SELinux) on Red Hat Enterprise Linux (RHEL) is an implementation of MAC that uses Linux Security Models that is based on the principle of least privilege. When enabled in permissive mode, every access to a system resource by a user or process such as an I/O device must be controlled by SELinux. This can sometimes cause extra processing cycles on the system.

AppArmor is a Linux application security framework that is included with SUSE Linux Enterprise and is an open source project. It takes a different approach from SELinux and provides an easy-to-use way for security applications in Linux. The following features are available in AppArmor:

- ▶ Yet another Setup Tool (YaST), which is an administration tool for configuration, maintenance, and automated development of a per-program security policy
- ▶ Predefined security policies for standard Linux programs and services
- ▶ Robust reporting and alerting capabilities to facilitate regulatory compliance
- ▶ Common Interface Model (CIM), which is a schema for clients that integrate with industry standard management consoles
- ▶ ZENworks Linux Management integration for profile distribution and report aggregation
- ▶ Path-name-based system that does not require labeling or relabeling of file systems

For more information about how to set up SELinux or AppArmor at Linux on z Systems, see *Security for Linux on System z*, SG24-7728.

8.3.3 User management

On a cloud environment, with the flexibility to increase and decrease its size, the number of user IDs and the complexity of managing the user IDs increase. It is important to have a way to manage user IDs, mainly from a security perspective.

Centralizing the repository of user IDs helps in the management activities, reducing the administration effort compared to distributed user IDs management. It is considered a preferred practice for the maintenance of the information security policies that are applied to user management.

The centralization of user ID management involves adding, deleting, changing account information, and resetting passwords. Doing that from a single and centralized point, such as an LDAP server, can help keep the security requirements and policies consistent throughout the cloud environment. This configuration avoids the need to spread sensitive information from users, such as passwords, to all servers.

When using a centralized user ID management server, all servers must connect to it by using an encrypted connection. Not all of the information flowing between the LDAP server and the servers on the cloud is sensitive, but enabling this protection is simpler to implement than using a mix of encrypted and non-encrypted connections.

8.3.4 Update management

Keeping the operating system updated helps prevent its exposure. An established update process under the servers tracks system updates and manages them in an acceptable time frame. Using a minimal system installation also helps keep control of security and system update. There are fewer potential points of security exposure when fewer packages are installed and managed.

The use of a centralized patch management tool can decrease the complexity and time spent to apply server patches when the number of servers being managed increases. It also helps to track patches that are applied and patches that are needed to all servers managed, avoiding the possibility to leaving a server without the updates.

8.3.5 Data

Protecting the access to the disks is important, but another way to improve data security is encrypting it. Because encryption depends on a key, correct handling and implementation of a key management policy is important. Failing on encryption key management can result in an encryption deadlock and permanent loss of all encrypted data.

The use of encryption on IBM LinuxONE has advantages because it uses cryptographic hardware and cryptographic functions that are built in the central processor, such as the Central Processor Assist for Cryptographic Function (CPACF). It handles the cryptographic cipher calculations, which leaves the central processor available for other uses and reduces the central processor cycles compared to the same cipher calculations that are done by using software emulation.

The use of tools to encrypt the Linux on IBM LinuxONE data can increase the data security. The dm-crypt subsystem in Linux is implemented as a device mapper that can be stacked on the top of other devices that are managed through the device mapper framework. Therefore, you can encrypt from entire disks to software RAID volumes and LVM logical volumes, adding flexibility to the encryption strategy. In a dm-crypt environment, the data appears in the clear only when it is already in the program.

For more information about how to encrypt data on disks by using dm-crypt, see *Security for Linux on System z*, SG24-7728.

Data on backup media must be encrypted. It is a security breach if a backup media with sensitive information leaves the data center and others outside the organization have access to that media. Protecting the data center and all devices within it is important, but allowing information to leave the data center without being protected is the same as ignoring all protection implemented in the organization data center.

8.3.6 Audit

A defined information security policy is worthless if there is no way to assess whether the policies are effective, meaning that it was adhered to by all employees and they are fulfilling the roles as expected.

Tracking changes, and authorized and unauthorized accesses, is a way to make sure that the information security policy is followed. But, with the increase in servers that are managed on the enterprise cloud, the amount of audit data that is generated makes it impossible for a human to analyze all of it, find a threat, and act on it while the intrusion is still happening. For that reason, define, during the planning stage of the enterprise cloud and the IT infrastructure, which actions must be logged for audits.

The complexity in auditing is reduced when defined roles are available in the information security policy. Users under one role should not have access to override the MACs and should not be able to manipulate the controls that are under the jurisdiction of another job role. With the separation of duties, the functions of the systems and integrity of audit logs are not compromised.

To create a separation of duties under Linux, use SELinux or AppArmor. If those tools are not used or enabled, the task to control and determine user privileges become more complicated. A preferred practice is to use **sudo** to control access to privileged commands. Use of **sudo** ensures better protection by limiting the privileged commands that a user can run and protecting the root password from being shared with system administrators. Use of **sudo** also ensures audit of accountability for users who run privileged commands.

8.3.7 Cryptographic hardware

Linux on IBM LinuxONE can benefit from the use of IBM LinuxONE cryptographic hardware. It supports the use of CPACF and Crypto-Express6S (the latest available at the time of this writing) by using in-kernel crypto-APIs and the libica cryptographic functions library. Use of these features provide the following benefits:

- ▶ File system encryption
- ▶ Communication encryption (to the applications such as IBM HTTP Server)
- ▶ System security by providing advanced cryptographic functions

CPACF is available on every processor unit that is defined as a central processor (CP) and can be explicitly enabled by using the enablement feature #3863, for no additional charge. It provides a set of symmetric cryptographic functions that enhance the encryption and decryption performance of clear-key operations for SSL, virtual private network (VPN), and data storing applications that do not require a high level of security.

The CPACF coprocessor on the IBM LinuxONE was redesigned and has better performance compared to previous servers.

CPACF offers the following data encryption and decryption algorithms for data privacy and confidentiality:

- ▶ Data Encryption Standard (DES):
 - Single-length key DES
 - Double-length key DES
 - Triple-length key DES (TDES)
- ▶ Advanced Encryption Standard (AES) for 128-bit, 192-bit, and 256-bit keys

CPACF offers the following hashing algorithms for data integrity:

- ▶ SHA-1: 160 bit
- ▶ SHA-2: 224, 256, 384, and 512 bit

For MAC, CPACF offers the following options:

- ▶ Single-length key MAC
- ▶ Double-length key MAC

For cryptographic key generation, CPACF offers Pseudorandom Number Generation (PRNG) algorithms.

Crypto-Express6S is a feature that was designed to complement the cryptographic capabilities of the CPACF. It is in the Peripheral Component Interconnect Express Generation 2 (PCIe Gen2) I/O drawer and can be configured in one of the following ways:

- ▶ Secure IBM Common Cryptographic Architecture (CCA) coprocessor (CEX6C), supporting:
 - Secure key functions
 - FIPS 140-2 Security Level 4 certification
 - User Defined Extension (UDX) services to implement custom cryptographic functions and algorithms

- ▶ Secure IBM Enterprise Public Key Cryptography Standards (PKCS) #11 (EP11) coprocessor (CEX6P) provides open, industry-standard cryptographic services following the PKCS #11 specification V2.20 and more recent amendments. It is designed to extend FIPS and Common Criteria evaluations to meet public sector requirements. The new cryptographic coprocessor mode introduced the PKCS #11 secure key function.
- ▶ Accelerator (CEX6A), which is optimized for public and private key cryptographic operations, is used with SSL/TLS processing.

The optional PCIe cryptographic coprocessor Crypto-Express6S provides asynchronous cryptographic functions to IBM LinuxONE servers. Over 300 cryptographic algorithms and modes are supported.

For more information about IBM LinuxONE servers, see this [web page](#).

For more information about how to use these features at Linux on z Systems, see *Security for Linux on System z*, SG24-7728.

8.3.8 Firewall

The use of a firewall is defined by the IT Infrastructure and by the information security policy of an organization. The use of the guest isolation feature under z/VM and IBM LinuxONE architecture makes such a solution as secure as having a firewall running on every Linux on IBM LinuxONE server. However, if the information security policy enforces the use of a firewall on any back-end server, including those running on an IBM LinuxONE environment, that firewall enforcement should be implemented.

Several sophisticated firewall feature solutions are available for Linux that can filter and manipulate packets based on complex rules that are defined by the system administrator. A preferred practice is to use a restrictive firewall policy instead of a permissive policy. This configuration ensures that packets that are explicitly not allowed to flow to the network are dropped instead of rejected.

The following tools help to automate firewall policy creation:

- ▶ SUSE Enterprise Linux offers a firewall configuration tool that uses YaST that can be used both in graphical mode or text mode.
- ▶ Red Hat Enterprise Linux offers a firewall configuration tool that is called system-config-firewall that can also be used in graphical or text mode.
- ▶ Ubuntu offers the Uncomplicated Firewall (UFW) as the default firewall configuration tool. Developed to ease iptables firewall configuration, UFW provides an easy way to create an IPv4 or IPv6 host-based firewall. By default, UFW is disabled.

Related publications

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

Other publications

The following publications are also relevant as further information sources:

- ▶ *Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283
- ▶ *Enabling z/VM for OpenStack (Support for OpenStack Newton Release)*, SC24-6253
- ▶ *IBM Cloud Manager with OpenStack on z Systems V4.2*, SC24-6251
- ▶ *RACF Security Server Security Administrator's Guide*, SC24-6311
- ▶ *RSCS Networking Planning and Configuration*, SC24-6320
- ▶ *Secure Configuration Guide*, SC24-6323
- ▶ *Systems Management Application Programming*, SC24-6327
- ▶ *The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3*, SG24-8147
- ▶ *The Virtualization Cookbook for IBM z Systems Volume 2: Red Hat Enterprise Linux 7.1 Servers*, SG24-8303
- ▶ *The Virtualization Cookbook for IBM z Systems Volume 3: SUSE Linux Enterprise Server 12*, SG24-8890
- ▶ *The Virtualization Cookbook for IBM z Systems Volume 4: Ubuntu Server 16.04*, SG24-8354

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

Securing Your Cloud: IBM Security for LinuxONE

SG24-8447-00
ISBN 0738457949



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



SG24-8447-00

ISBN 0738457949

Printed in U.S.A.

Get connected

