Redbooks

ibm.com/redbooks

IBM LinuxONE Resiliency

Ewerson Palacio

Brian Hugenbruch

Wilhelm Mild

Filipe Miranda

Anderson Augusto Silveira

Livio Sousa

Bill White



IBM LinuxONE

IBM

Redbooks



IBM Redbooks

IBM LinuxONE Resiliency

September 2024

SG24-8544-00

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

First Edition (September 2024)

This edition applies to IBM LinuxONE Emperor, LinuxONE Rockhopper 4, and IBM z16.

© Copyright International Business Machines Corporation 2024. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices vii
Trademarks viii

Preface ix
Authors ix
Now you can become a published author, too! x
Comments welcome x
Stay connected to IBM Redbooks x

Chapter 1. IT resiliency matters 1
1.1 The impact of downtime 2
1.1.1 Outages do happen 2
1.2 IT resiliency is a journey 3
1.2.1 Reducing risk and impact on service delivery 3
1.2.2 Service-level agreements drive resiliency requirements 3
1.2.3 SLA criteria dictate requirements for recovery 4
1.3 IT resiliency goes beyond disaster recovery 5
1.3.1 IT resiliency challenges 5
1.4 How to prepare for IT resiliency 6
1.4.1 Balancing the risks and costs of mitigation 7
1.4.2 How IBM LinuxONE helps 8
1.5 Deployment models for IBM LinuxONE resiliency 9
1.5.1 IBM LinuxONE deployment models 9

Chapter 2. Technical view on IBM LinuxONE resiliency 15
2.1 Layers of resiliency based on IBM LinuxONE 16
2.2 Infrastructure 16
2.2.1 IBM LinuxONE built-in hardware resiliency 17
2.2.2 Physical storage attachment capabilities 22
2.2.3 Physical network attachment capabilities 24
2.3 Operating environments 25
2.3.1 Virtualization 25
2.3.2 IBM Processor Resource Systems Manager 26
2.3.3 Dynamic Partition Manager 28
2.3.4 Highest workload isolation in LPAR 30
2.3.5 The z/VM hypervisor on IBM LinuxONE 30
2.3.6 The KVM hypervisor on IBM LinuxONE 33
2.3.7 Container platforms 34
2.4 LinuxONE virtual storage 35
2.5 LinuxONE Virtual Network 37
2.5.1 IBM HiperSockets 37
2.5.2 KVM MacVTap 38
2.5.3 z/VM Virtual Switch 39
2.5.4 HyperSwap 39
2.5.5 Multipathing 40
2.6 Applications 41
2.6.1 Containers 41
2.6.2 Containerization 42
2.6.3 Containerized workloads 42

© Copyright IBM Corp. 2024.

2.6.4 Middleware 42
2.6.5 Independent software vendor applications 44
2.6.6 Databases 44
2.7 Management 45
2.7.1 z/VM Hypervisor Management 45
2.7.2 KVM Hypervisor Management 46
2.7.3 Container management 46
2.7.4 Kubernetes 46
2.7.5 Red Hat OpenShift Container Platform 47
2.8 Automation 47
2.8.1 Synchronous data replication 48
2.8.2 Copy Services Manager 48
2.8.3 Automation with Linux HA components 48
2.8.4 Automation with Ansible: Overview 50
2.8.5 IBM Cloud Infrastructure Center automation 51
2.8.6 Resiliency automation for the infrastructure 51
2.8.7 Automation with GDPS 51
2.8.8 Infrastructure placement 57
2.9 Building the related deployment model 58
2.9.1 Best practices for hardware and virtualization 58
2.9.2 Workload availability 59

Chapter 3. Deployment scenarios for resiliency 61
3.1 LinuxONE single-site scenarios 62
3.1.1 Reliable base 62
3.1.2 Flexible site 63
3.2 Multi-site scenario 68
3.2.1 Multi-site scenario details 68
3.2.2 Applicable GDPS capabilities for a multi-site scenario 69
3.2.3 Multi-site possible configurations 70
3.2.4 Implementing z/VM Single System Image 74
3.2.5 HyperSwap function 74
3.2.6 PAV, HyperPAV, and HyperSwap in z/VM 75
3.2.7 Resiliency level of multi-site scenarios 75
3.3 Continuous availability scenario 76
3.3.1 Continuous availability scenario details 76
3.3.2 Continuous availability scenario environment 76
3.3.3 GDPS Continuous Availability capabilities 77
3.3.4 Resiliency level of continuous availability 79
3.3.5 “Eight nines” (99.999999%) availability 80

Chapter 4. Red Hat OpenShift Container Platform resiliency 81
4.1 Red Hat OpenShift Container Platform resiliency 82
4.1.1 Calculating availability 82
4.1.2 Differentiators 83
4.2 Availability and resiliency for Linux workloads 84
4.2.1 Components of LinuxONE resiliency 84
4.2.2 Network 86
4.2.3 Network optimization 86
4.2.4 Storage optimization 87
4.3 Framework analysis 89
4.3.1 Points of outage 89
4.3.2 Planned outages 103

4.4 Disaster recovery 106
4.5 Resources 109

Related publications 111
IBM Redbooks 111
Online resources 111
Help from IBM 111

Abbreviations and acronyms 113

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.

© Copyright IBM Corp. 2024.

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 https://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.

Db2®	IBM Z®	Resource Link®
DS8000®	IBM z13®	Tivoli®
FICON®	IBM z14®	z/Architecture®
FlashCopy®	IBM z16™	z/OS®
GDPS®	Interconnect®	z/VM®
HyperSwap®	OMEGAMON®	z13®
IBM®	Parallel Sysplex®	z15®
IBM Cloud®	Redbooks®	z16™
IBM Spectrum®	Redbooks (logo)  ®	

The following terms are trademarks of other companies:

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

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

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

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

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

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

Preface

This IBM Redbooks® publication describes the concepts of resiliency as they relate to information technology systems. This book focuses on a LinuxONE server as a foundation model. Understanding uptime helps you understand the importance of availability for a business, institution, or governmental agency.

Authors

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

Ewerson Palacio is an IBM Redbooks Project Leader. He holds a bachelor's degree in Math and Computer Science. Ewerson worked for IBM Brazil for over 40 years and retired in 2017 as an IBM Distinguished Engineer. Ewerson has co-authored many IBM Z® publications, and created and presented IBM Redbooks seminars worldwide.

Brian Hugenbruch CISSP is a Senior Software Engineer with IBM Z and LinuxONE. Officially, he is the IBM z/VM® Cryptography and Security Development lead. He has also served as the LinuxONE Resiliency Technical Lead since 2020, and in this capacity he drove the platform to its "Eight 9s" calculation. He combines these two sides of his work in the topics of cyber resiliency and digital forensics. He writes and speaks on these topics in numerous forums and conferences.

Wilhelm Mild is an IBM Executive IT Architect and Open Group Distinguished Architect. He designs IBM LinuxONE and IBM Z solution architectures that feature containerization, Red Hat OpenShift Container Platform, end-to-end security, resiliency, high availability and disaster recovery (HADR), and network topologies for complex application landscapes. He speaks at international customer events, and teaches classes and hands-on labs.

Filipe Miranda is a principal technical specialist for the Red Hat Synergy team at IBM®. He has 15+ years of experience with open-source technologies. He is an author of several IBM Redbooks and technical articles. He is a member of the zAcceleration team. His key skills include IBM Cloud® Paks, Red Hat OpenShift, DevOps, and other technologies that are part of the hybrid cloud portfolio.

Anderson Augusto Silveira is a Senior Technical Specialist for LinuxONE, Linux on IBM Z, and Red Hat OpenShift within IBM. He has 12+ years of experience in virtualization and open-source technologies. He works on mission-critical projects worldwide, trains teams of IBM Business Partners and Clients, and helps and supports the adoption of new technologies and products.

Livio Sousa is an IT Specialist with over 20 years of experience with high-end platforms, such as servers, storage, networking equipment, and enterprise architecture integration. He supports hybrid cloud solutions. He has worked with IBM z/OS®, Linux, UNIX, and Windows, and different hardware processors architectures like IBM z/Architecture®, IBM Power, and the IA-32 architecture and its extensions.

Bill White is a Project Leader and Senior Infrastructure Specialist at IBM Redbooks, Poughkeepsie Center.

© Copyright IBM Corp. 2024.

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

Steven Cook and Justin VanSlocum
IBM US

Now you can become a published author, too!

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

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

`ibm.com/redbooks/residencies.html`

Comments welcome

Your comments are important to us!

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

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

  `ibm.com/redbooks`

- Send your comments in an email to:

  `redbooks@us.ibm.com`

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on LinkedIn:

  `https://www.linkedin.com/groups/2130806`

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

  `https://www.redbooks.ibm.com/subscribe`

- Stay current on recent Redbooks publications with RSS Feeds:

  `https://www.redbooks.ibm.com/rss.html`

1

IT resiliency matters

IT resiliency is about the entire IT infrastructure being consistently available, which includes services and data. Also, many organizations operate in highly regulated industries and must comply with strict IT resiliency standards. Service providers offer certain service levels pertaining to quality, availability, and responsiveness that they are obligated to meet.

This chapter describes the concepts of resiliency as they relate to information technology systems. This book focuses on a LinuxONE server as a foundation model. Understanding uptime helps you understand the importance of high availability (HA), business continuity (BC), and disaster recovery (DR).

This chapter includes the following topics:

- 1.1, “The impact of downtime” on page 2
- 1.2, “IT resiliency is a journey” on page 3
- 1.3, “IT resiliency goes beyond disaster recovery” on page 5
- 1.4, “How to prepare for IT resiliency” on page 6
- 1.5, “Deployment models for IBM LinuxONE resiliency” on page 9

© Copyright IBM Corp. 2024.

1.1 The impact of downtime

There is no room for downtime in today's digital world. Consumers of any service, such as someone who uses an app on their phone, or application developers or business partners relying on infrastructure, expect services to be available. Services must be available without disruption.

There is a cost when those expectations are not met, and that cost can be significant: legal consequences, lost revenue, damaged reputation, stalled productivity, or failure to meet regulatory and compliance standards.

The impact of downtime can vary greatly depending on the business. Some businesses might be able to survive an outage of a few hours, but for other businesses, even consumers experiencing poor performance for several seconds can impact the bottom line. Not being able to deliver services as expected or at the agreed level can potentially turn into a loss of business or hefty fines.

IT resiliency must protect your business from downtime. IT infrastructure solutions must work together to optimize availability, keep your systems running, detect problems in advance, and recover your critical data.

1.1.1 Outages do happen

Outages are the result of planned and unplanned events.

A *planned outage* is an anticipated and controlled disruption of a service or services. It is a scheduled event where consumers are notified in advance with a description and scope of the outage with a specific date, time, and duration. Reasons for a planned outage can include testing or implementing new hardware or software components, maintenance to apply fixes, or configuration changes that require a system restart. This can involve a partial or a full site outage.

An *unplanned outage* is an unforeseen disruption to a service or services, when one or more hardware or software components of the service fails. Reasons for an unplanned outage can be caused by a natural disaster, cyberattacks, infrastructure failure, or human error.[1]

The best approach to remedying any type of outage is to eliminate single points of failure (SPOFs) by creating an environment that has redundant resources with an automated and seamless takeover process. Without such measures, you must first identify and fix the problem, and then manually restart the affected hardware and software components.

Even planned outages for businesses with 24x7 service offerings can come at a cost. Businesses must be ready and able to recover as quickly and seamlessly as possible from any type of event to minimize cost.

For more information about the cost of downtime, see The Real Costs of Planned And Unplanned Downtime.

[1] The global average total cost of a data breach is $4.45 million. For more information, see “Cost of a data breach 2023” at https://www.ibm.com/reports/data-breach.

1.2 IT resiliency is a journey

IBM defines IT resiliency as the ability to rapidly adapt and respond to any internal or external opportunity, demand, disruption, or threat, and continue business operations without significant impact. This includes continuous and near-continuous application availability, keeping your applications running throughout the far more common planned and unplanned outages, and DR, which concentrates on recovering from an unplanned event.

1.2.1 Reducing risk and impact on service delivery

Resiliency refers to a component's ability to deliver the intended outcome despite any type of failure or abnormal conditions. It is not only helping ensure that the hardware is running. System resiliency applies to the entire stack (compute, operating system (OS), workload, network, and storage layers). It should consistently be available to help ensure that services and data are accessible. In addition, transactions should complete successfully and on time, with good performance.

The entire IT infrastructure might need to adhere to regulatory guidelines and standard rules, which also impact system resiliency. For example:

- NIST SP 800-160 and NIST SP 800-193
- PCI DSS 9.5.1
- US White House Executive Order 14028, "Improving the Nation's Cybersecurity"
- European Union Digital Operational Resilience Act (DORA)[2]
- ISO 27001 (Information Security Management)

Because of its checklist-oriented nature, many businesses and organizations use the Payment Card Industry Data Security Standard (PCI DSS) as a starting point for building system resilience. Then, they might apply stronger standards when they are either needed for the client experience, or they are required by law.

Regulations and standards that apply to various industries are frequently updated to reflect what is happening in the cyber resiliency aspect of their industries. For multinational businesses, the laws, policies, and regulations might differ in various geographic locations, which cause more stress on the requirements for IT infrastructure resiliency.

The design of systems that support critical IT services depends on the interaction between the criticality of the service and its business profile. It is a journey toward reducing risk and impact on service delivery. It becomes important to address downtime from a business viewpoint that is directed by service-level agreements (SLAs).

1.2.2 Service-level agreements drive resiliency requirements

For any service provider, SLAs determine how their IT infrastructure is designed, configured, and managed to provide the required level of resiliency. This is true whether you are discussing a service as wide-ranging as cloud-hosted platforms, or as small as an application that is shared within your IT department.

SLAs are used between a service provider and consumers to describe the service and define the level and quality that can be expected. It includes the duties and responsibilities of each party, the *metrics* for measuring service quality, and the course of action or penalties if the agreed service levels are not met.

[2] The Digital Operational Resilience Act (DORA) - Regulation (EU) 2022/2554

SLAs also specify terms and conditions, for example:

- Effective start dates, end dates, and review dates.
- Key performance indicators that track performance over time, for example:
  - Service ABC must be able to run at least 10,000 transactions/second with 95% finishing under 0.1-seconds response time as measured by application records.
  - Service ABC must be available 24/7, except for a 1-hour planned outage, once per quarter. Warning of this planned outage must be given no later than 2 weeks before the outage. The availability level is based on the value that is indicated in the outage record and agreed to by all parties.
- Commitments that are related to the type of record that the SLA applies to. For example, problem tickets specify target dates and times for response, resolution, delivery, availability, and other values. Work orders specify target start, finish, and delivery times.
- Other criteria that might be measured include defect rates, technical quality, security controls, and compliance standards.

1.2.3 SLA criteria dictate requirements for recovery

Resiliency requirements should reflect SLAs and their recovery objectives, which in turn drive the technology that is needed to satisfy them. It should not be the technology driving the resiliency requirements.

Recovery events are typically measured by the targets that are defined for recovery point objective (RPO) and recovery time objective (RTO) targets.

Looking backward from an event, the RPO represents how much data is lost due to a failure. This value can vary from zero or near-zero with synchronous remote copy to hours or days with recovery from tape. Looking forward from the event, the RTO is the goal for how much time can pass before service is restored. It includes starting the OS on the recovery system, the workloads, management tools, enabling the network, and bringing the services online. This time span can be anywhere from minutes to days, depending on the recovery solution.

It seems obvious to say that you must choose the replication techniques that achieve the recovery objectives. However, business realities often complicate the choices, for example:

- An existing building is owned at a remote location so it must be used, even though it is too far away to support synchronous replication to get RPO=0.
- Tape-tape replication is chosen because it is cheaper, even though the amount of potential data loss might impact business results.
- The SLA, RPO, and RTO were defined when the environment was different and are now outdated. Regular reviews and updates of existing SLAs, RPO, and RTO are needed to capture changes to the business environment.

Also, if there are problems with the recovery process there are dependencies on the infrastructure support staff. Automation can help in greatly reducing the time that is needed to recover the environment by optimizing the recovery process.

1.3 IT resiliency goes beyond disaster recovery

IT resiliency is related to, but broader in scope, than DR. DR concentrates solely on recovering from an unplanned event.

IT resiliency helps ensure continuity of service and support for its consumers and maintain its viability before, after, and during an event. A business must use of people, processes, and technology:

- People who understand their roles and responsibilities during an outage or disruption. They must design the strategies and solutions to enable the continuity of business operations. Operators must practice *recovery procedures* to help ensure a smooth recovery after an outage or disruption.
- Processes include crisis management and BC, but can be expanded to include problem, change-and-configuration management, and business controls.
- Technologies are often emphasized too much. However, they include backup, replication, mirroring, and failover solutions.

1.3.1 IT resiliency challenges

The challenges when achieving IT resiliency include the following items:

- Evaluating, planning, and mitigating the business impact of various types of risks.
- Responding to regulatory pressures and compliance that are associated with business resiliency.
- Helping ensure that data is protected, available, and accessible as needed by the business.
- Achieving availability objectives by reducing the frequency and duration of infrastructure applications and data outages.
- Recovering from and responding to disruptive events.

Several terms are used in for different levels or types of availability: HA, continuous operations (CO), continuous availability (CA), DR, and BC.

High availability

A HA environment has a fault-tolerant and failure-resistant infrastructure that supports continuous workload processing. The goal is to provide service during defined periods, at acceptable or agreed on levels, and mask *unplanned outages* from consumers. Many hardware and software components have built-in redundancy and recovery routines to mask possible events. The extent of this support varies greatly.

HA refers to systems and services that typically operate 24x7 and usually require redundancy and rapid failover capabilities. These capabilities minimize the risk of downtime due to failures in the IT infrastructure.

HA is distinguished from DR primarily through the duration and scope of outage that the technology is designed to mitigate.

Continuous operations

A CO environment enables nondisruptive backups and system maintenance that is coupled with CA of services. The goal of BC is to mitigate or mask the effect of planned outages from consumers. It employs nondisruptive hardware and software changes, nondisruptive configuration, and software coexistence. z/VM Live Guest Relocation (LGR), for example, enables CO because workloads can be moved from one system to another system at will.

Continuous availability

CA is a combination of CO and HA. It is designed to deliver nondisruptive service to consumers 24/7. There is no impact from planned or unplanned outages. Red Hat OpenShift Container Platform is an example of a CA solution. It enables HA by removing SPOFs within the hardware and its software environment. It enables CO by enabling “rolling system restarts”, nondisruptive updates, and workload relocation to mask planned events.

Disaster recovery

DR is protection against unplanned outages through a reliable and predictable recovery process. Examples include the inability to access storage, or a site disaster.

DR is the ability to respond to an interruption in services by implementing a DR plan to restore critical business functions. DR is more reactive than HA solutions. DR solutions typically enable recovery within 24 - 72 hours, and HA solutions enable recovery within 1 - 2 hours of an event. As with any event, the duration and scope of the event dictates when usage of the DR plan is necessary. A DR process might be an appropriate response, even with events that last less than the 24 - 72-hour standard.

Business continuity

BC is the ability of a business to withstand outages and operate important services normally and without interruption in accordance with SLAs.

BC includes high availability and disaster recovery (HADR), and it can be defined as the ability to withstand all outages (planned, unplanned, and disasters) and provide continuous processing for all critical workloads. The goal is for the outage time to be less than 0.001% of total service time. A HA environment typically includes more demanding RTOs (seconds to minutes) and more demanding RPOs (zero user disruption) than a DR scenario.

HA solutions provide a fully automated failover to a backup system so that users and applications can continue working without disruption. HA solutions must be able to provide an immediate recovery point. At the same time, they must provide a recovery time capability that is better than the recovery time that you experience in a non-HA solution topology.

1.4 How to prepare for IT resiliency

It can seem like a daunting challenge to create an always-on service, and there is no one technology that solves all problems. IT resiliency is a choice. So, what should be considered when tackling this problem?

First, consider points of failure. Any component of your system that does not have a backup can lead either to reduced throughput, reduced processing power, or a service outage.

Here are some basic assumptions for building a resilient IT system:

- Build the approach, and then test the approach. Include the following items:
  - Redundancy.
  - Early detection.
  - Automation and failover.
  - Test, practice, and improve. It is not a plan if you never tried it. It is only an idea.
- Resiliency best practices (generally speaking).
- How do these things address the concerns?
  - Financial: Cost-benefit analysis of downtime versus technical investment.
  - Regulatory: Cost of failing to comply.

To solve these challenges, invest in technologies that help reduce all downtime.

1.4.1 Balancing the risks and costs of mitigation

Higher availability comes at a higher cost. The cost varies depending on which services are affected. You can calculate the cost from consider the following aspects:

- Number of consumers that are affected by the outage
- Duration of the outage from the consumers' perspective
- Average number of transactions that run per hour per consumer
- Estimated revenue per transaction

Although such calculations are rough estimates, they can provide an investment baseline on the risk of investing in resiliency technologies or trusting to luck or fate.

Such calculations are done by using relative percentiles of availability, versus the cost of an outage. Having a 99% available system means that, in a year, you might anticipate 3.65 days of downtime. If one measures the cost of downtime by the hour, this can still lead to an expensive cost, even if the outage is planned.

In comparison, a system with seven 9s of availability (99.99999% uptime) is, in a year, unavailable for approximately 3.16 seconds, and the cost of this outage is significantly lower (see Table 1-1).

Table 1-1 Cost of outages in US$

Cost	99%	99.9%	99.95%	99.99%	99.99999%
$150,000+ / hour (98% respondents)	$8,766,000	$876,600	$438,300	$87,660	$88
$300,000+ / hour (88% respondents)	$26,000,000	$2,640,000	$1,315,000	$262,300	$263
$1,000,000+ / hour (40% respondents)	$87,660,000	$8,766,000	$4,383,000	$876,600	$877

Outages are never precisely as short or as long as shown in Table 1-1, and these estimates are based on a mean-time between failures, that is, how likely, in instances per year or per decade, they are likely to occur.

Balancing the risks and costs of mitigation is essential to achieving business goals. By balancing the risk of a type of failure and the cost of a solution, you can determine how much redundancy and availability management you can design into your IT infrastructure. By understanding the real cost of service unavailability, you can calculate the costs that you can sustain to provide this availability. These factors also determine when you can do maintenance and upgrades to the IT infrastructure.

As illustrated in Figure 1-1, you must evaluate the risk and impact of the outage and decide whether the design must be changed to meet the cost implications of service outages. Maybe these cost implications should be adjusted?

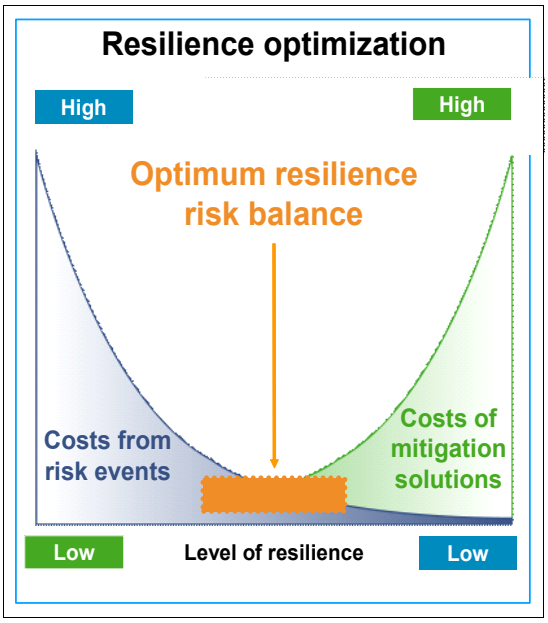


Figure 1-1 Resiliency optimization

Under-investment can result in “spending money by not-spending money,” increasing risks to the business. If an outage occurs, you might be able to bear the cost of recovery at the time of the outage, but the cost of recovery can be greater than the cost of prevention. Over-investment can result in excess solution costs. By overspending on high levels of protection, you can incur expenses and risks that are associated with lost opportunity.

1.4.2 How IBM LinuxONE helps

IBM LinuxONE has a sterling record of system uptime. The relative availability of the underlying server hardware, through a host of technical means that are described in Chapter 2, “Technical view on IBM LinuxONE resiliency” on page 15, can boost the uptime of applications by virtue of the system on which it is running.

This is not to say that one server is enough, or that LinuxONE hardware solves all problems. Again, system resiliency is a lifestyle choice, and this means that every SPOF in storage, compute, networking, virtual infrastructure, physical site maintenance, and others must be addressed as part of an overall resiliency and DR plan.

1.5 Deployment models for IBM LinuxONE resiliency

The goal of a deployment model is to provide a degree of horizontal scaling: If one copy of the workload fails, there is another copy to take its place.

For the purposes of this book, deployment models are frameworks for planning and implementing resiliency in an IBM LinuxONE environment. The focus of these models is less about workload configuration and more about the level of resources that are needed in a particular part of your IT environment.

For more information about the IBM LinuxONE deployment models by using different use case scenarios with architectural patterns, see Chapter 3, “Deployment scenarios for resiliency” on page 61.

An *architectural pattern* is a specific selection of hardware and software that is meant to illustrate in detail how the various parts of an infrastructure can work together to anticipate unplanned outages, mitigate planned outages, or help in times of disaster.

The goal of an architectural pattern is to provide vertical scaling for resiliency, that is, in a stack, how can the parts help offset one another? Architectural patterns are constructed of the components that are described in Chapter 2, “Technical view on IBM LinuxONE resiliency” on page 15, and an illustration of both a deployment model and architectural pattern is provided in Chapter 4, “Red Hat OpenShift Container Platform resiliency” on page 81.

1.5.1 IBM LinuxONE deployment models

To help with improving the resiliency of an IBM LinuxONE environment, here are a series of deployment models for resiliency. Those models map to different levels of resiliency based on common IBM LinuxONE configurations and specific business requirements. The models can guide you through an assessment of your current resiliency investment and select new targets for workloads. The models can help to identify practical steps to achieve availability goals for your IT infrastructure. In addition, the models suggest DR considerations between data centers as part of your assessment.

Table 1-2 provides an overview of the four IBM LinuxONE resiliency deployment models.

Table 1-2 Deployment models for IBM LinuxONE resiliency

Model number	Model name	Risk/impact to business	Time to recovery	Typical RTO[a]
1	Reliable base	High	8+ hours	3+ day
2	Flexible site	Medium	2+ hours	8 hours
3	Multi-site resiliency	Low	Minutes	< 1 hour
4	CA	Minimum	Seconds	< 120 seconds

a. RTO is the target time for system restoration, and RPO is the data loss that is acceptable. (The RPO could be zero data loss.)

Each of the four models supports different capabilities of the hardware and software technologies that are represented by the layers that are listed in Table 1-3. The table also emphasizes that the layers relate to each of the four resiliency deployment models.

Table 1-3 Deployment models layers

Layer	Components examples of each layer
Infrastructure	Compute (CPUs, memory, and firmware), storage, and network
Operating environments	Virtualization (IBM z/VM, Kernel-based Virtual Machine (KVM), IBM PR/SM, Dynamic Partition Manager (DPM)), Linux, and container platforms
Applications	Containerized workloads, middleware, independent software vendor (ISV) applications, and databases
Management and automation	► Hypervisors' management ► Container management ► Data replication

All deployment models in Table 1-2 on page 9 have a relationship with the aspects of all layers that are shown in Table 1-3.

In the relationship of the layers to the four resiliency deployment models, the question is this: What are the essential features of the layers for each of the four models? This topic is described in Chapter 2, "Technical view on IBM LinuxONE resiliency" on page 15.

Here are some example issues regarding layers in relation to the four models:

- Model 1 to Model 2: In the transition from resiliency model 1 to 2, the hardware layer typically grows from a single system to a multi-system environment.
- Reference architectures: Various IBM solutions can be a part of your implementation of a model. Each solution has a *reference architecture*, which can include hardware components and software.

For more information about the capabilities of each layer, see Chapter 2, "Technical view on IBM LinuxONE resiliency" on page 15.

The selection of a deployment model should be based on a combination of risk and cost. If a smaller business, for example, has only one data center, then this business is constrained by the limits of physical site construction as to the number of threats that can be defeated.

Deployment model 1: Reliable base

This deployment model is designed as a "starter" for resiliency, with a focus on vertical scaling. It contains one physical site and one IBM LinuxONE system.

It would be best never to have failures. You can get closer to this state by starting with quality base technology together with regular maintenance. Application code that you write on top of that base should have extensive problem detection and near-instantaneous correction. You can take these steps with IBM LinuxONE hardware and the OS.

By design, the IBM LinuxONE platform is a highly resilient system. Its architecture has built-in self-detection, error correction, and redundancy. The IBM LinuxONE reduces SPOFs to deliver the best reliability of any enterprise system in the industry. Transparent processor sparing and dynamic memory sparing enable concurrent maintenance and seamless scaling without downtime. Workloads that fail can be restarted in place.

If the OS or logical partition (LPAR)[3] is down for a planned or unplanned outage, another instance of the workload (in another LPAR) can absorb the work. An automatic recovery process can reduce the workload impact by avoiding the delays that are associated with a manual intervention.

Resiliency for a single IBM LinuxONE platform can be enhanced by using a clustering technology with data sharing across two or more Linux images. Although this does not provide for extra hardware redundancy, it can keep workloads available and accessible across software upgrades by allowing rolling restarts.

For more information about this model, see 3.1.1, “Reliable base” on page 62.

Deployment model 2: Flexible site

This deployment model expands on the reliable base model by adding redundancy for compute, storage, and network within a single site for greater control of outages.

Despite the best base technology, failures can still occur. Anticipate failures with the right technology implementation. The flexible site model can restart workloads quickly on a backup component, which can significantly reduce the impact of an outage.

With the installation of a second IBM LinuxONE system in the IT infrastructure, your component resource sharing begins the road to full system redundancy. Also, SPOFs for data can be mitigated by setting up Peer-to-Peer Remote Copy (PPRC), Metro Mirror (MM) (synchronous) if the systems are installed separated by a metro distance (up to 300 km[4]), and Global Mirror (GM) (asynchronous) for longer distances. IBM Copy Services Manager (CSM) can enable fast data replication with little or no data loss. For more information, see 2.8.8, “Infrastructure placement” on page 57.

For more information about this deployment model, see 3.1.2, “Flexible site” on page 63.

Deployment model 3: Multi-site resiliency

This deployment model is sometimes achieved without going through deployment model 2. It introduces one IBM LinuxONE system and its associated storage into a second data center, which removes the limitation of physical site recovery from SPOFs. This step is necessary on the journey to higher availability and DR.

Even the best fault-tolerant designs can fail. When failure happens and service is impacted, be prepared to restore service quickly. One method to reduce the impact of an outage is to avoid outages whenever possible. Anticipate failures with the right technology implementation.

Most hardware and software components in the IBM LinuxONE system can be cloned to a second IBM LinuxONE system with dynamic workload balancing spreading the work. For planned or unplanned events on one LPAR, the workload flows to another LPAR seamlessly, without affecting availability.

By adding IBM GDPS® capabilities to the disk replication, you can automate management of system and site recovery actions. In failover situations, automation can reduce the business impact by minutes or even hours.

---

[3] A subset of the IBM LinuxONE hardware that is defined to support an OS. An LPAR contains resources (such as processing units (PUs), memory, and input/output interfaces) and operates as an independent system. Multiple LPARs can exist in an IBM LinuxONE platform. For more information, see 2.8.3, “Automation with Linux HA components” on page 48.

[4] IBM Storage DS8870 supports up to 300 km (186 mi) with MM. Delays in response times are proportional to the distance between volumes.

For more information about this deployment model, see 3.2, “Multi-site scenario” on page 68.

With multiple systems, multiple storage units, and automation software at multiple data centers, every possible risk can be addressed before it happens. The distance at which physical sites are set also becomes a defining factor because two data centers at a metro proximity might fall subject to the same disasters.

Each step of the journey involves time and money to procure, install, configure, and test each of the components that are involved. Testing is a requirement because investing in these components without validating that they work might lead to unwanted surprises in an actual disaster.

Deployment model 4: Continuous availability

The LinuxONE continuous availability deployment model requires that you implement strategies and technologies that help ensure uninterrupted and reliable operations on IBM LinuxONE systems.

GDPS for Continuous Availability

GDPS is used to provide CA and DR for IBM Z, including LinuxONE. It orchestrates workload routing and facilitates failover during disruptions, which helps ensure uninterrupted operations.

To implement LinuxONE continuous availability, z/OS[5] partitions running on IBM Z in a GDPS environment are required. The GDPS Extended Disaster Recovery (xDR) function extends the near-CA and DR capabilities that are provided by GDPS Metro to other platforms or OSs that run on LinuxONE servers.

GDPS Metro Multiplatform Resiliency for IBM Z

GDPS Metro provides a function that is called GDPS Metro Multiplatform Resiliency for IBM Z (xDR). This function is especially valuable for customers who share data and storage subsystems between z/OS and Linux z/VM guests on IBM Z, for example, an application server running on Linux on IBM Z and a database server running on z/OS.

With a multi-tiered architecture, you must provide a coordinated and near-continuous availability and disaster recovery (CADR) solution for both z/OS and Linux. GDPS Metro can provide this capability when Linux is running as a z/VM guest or native on LinuxONE or IBM Z. You can use the IBM HyperSwap® function so that the virtual device that is associated with one real disk can be swapped transparently to another disk. HyperSwap can switch to secondary disk storage subsystems that are mirrored by MM. If there is a hard failure of a storage device, GDPS coordinates the HyperSwap with z/OS for CA spanning the multi-tiered application.

For site failures, GDPS starts the Freeze function for data consistency and rapid application restart so that there is no need for data recovery. HyperSwap can also be helpful in data migration scenarios to enable applications to migrate to new disk volumes without requiring them to be quiesced.

When using an Extended Count Key Data (ECKD) formatted disk, GDPS Metro can provide the reconfiguration capabilities for the Linux servers and data in the same manner as for z/OS systems and data. To support planned and unplanned outages, these functions are extended to KVM on LinuxONE and IBM Z with GDPS 4.1.

[5] LinuxONE does not support or run z/OS.

GDPS provides the following recovery actions:

- Restarts failing OS images in place.
- z/VM LAN management.
- Manages z/VM LPARs and z/VM guests, including Linux on IBM Z.
- Heartbeat checking of Linux guests.
- Disk error detection.
- Data consistency with freeze functions across z/OS and Linux.
- Site takeover/failover of a complete production site.
- Single point of control to manage disk mirroring configurations.
- Coordinates recovery for planned and unplanned events.

More support is available for Linux running as a guest under z/VM:

- Restarts failing OS images in place.
- Performs an orderly Linux node or cluster startup and shutdown.
- Coordinates planned and Unplanned HyperSwap (UHS) of disk subsystems, which is transparent to the OS images and applications that use the disks.
- Performs transparent disk maintenance and failure recovery with HyperSwap across z/OS and Linux applications.

The Continuous Availability Deployment model that is described in 3.3, “Continuous availability scenario” on page 76 is based on a configuration running z/OS images on IBM Z, z/VM, KVM, and Linux on IBM Z LPARS running on LinuxONE.

2

Technical view on IBM LinuxONE resiliency

This chapter explores IBM LinuxONE capabilities to help achieve the highest platform resiliency and availability.

It describes the various aspects of a virtualized environment, including the IBM LinuxONE processor reliability, availability, and serviceability (RAS) characteristics; Processor Resource System Manager (PR/SM) and Dynamic Partition Manager (DPM) operation modes; and virtualization and hypervisors, such as z/VM and Kernel-based Virtual Machine (KVM). It also covers resiliency automation for the infrastructure; the LinuxONE Input/Output (I/O) capabilities with Fibre Channel connection (IBM FICON®) and Fibre Channel Protocol (FCP) SCSI; and networking through internal network capabilities and network cards.

This chapter includes the following topics:

- 2.1, “Layers of resiliency based on IBM LinuxONE” on page 16
- 2.2, “Infrastructure” on page 16
- 2.3, “Operating environments” on page 25
- 2.4, “LinuxONE virtual storage” on page 35
- 2.5, “LinuxONE Virtual Network” on page 37
- 2.6, “Applications” on page 41
- 2.7, “Management” on page 45
- 2.8, “Automation” on page 47
- 2.9, “Building the related deployment model” on page 58

© Copyright IBM Corp. 2024.

2.1 Layers of resiliency based on IBM LinuxONE

To build an environment for highest resiliency on IBM LinuxONE machines, consider all the layers that are involved and might impact the resiliency of a running environment as shown in Figure 2-1. Also, consider which layers might be in a reliable environment and help provide the best availability of the application services, regardless of planned or unplanned outages of the components on any layers that are part of the environment.

Figure 2-1 illustrates these layers. It shows options on the right that might contribute to building an environment with high resiliency with IBM LinuxONE.

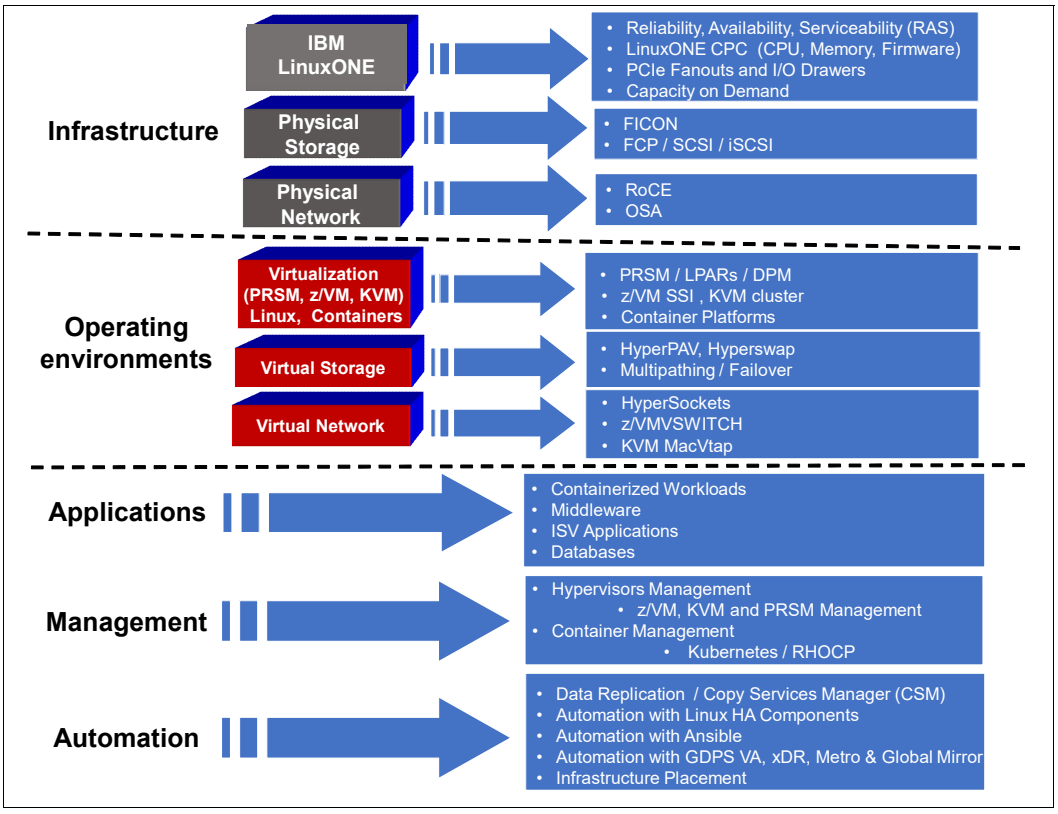


Figure 2-1 LinuxONE resiliency layers

2.2 Infrastructure

This section describes the infrastructure layer, which includes the IBM LinuxONE built-in hardware resiliency components and its RAS platform enhancements. It also covers the IBM LinuxONE central processor complex (CPC) components and the infrastructure layer's key capabilities, including compute, physical storage, and the physical network.

2.2.1 IBM LinuxONE built-in hardware resiliency

The IBM LinuxONE system helps support mission-critical applications for the most diverse industries and delivers exceptional overall resiliency. Resiliency is accomplished with built-in redundancy, simultaneous replacement, repair, and upgrade functions for the CPC, processing units (PUs), memory and I/O drawers, and I/O components for storage and networking. IBM LinuxONE resiliency also extends the capability to install firmware updates (Licensed Internal Code) nondisruptively.

Usually, LinuxONE capacity upgrades can be concurrent and without a system restart. For more information, see “Capacity on Demand (temporary upgrades)” on page 22.

Resilience capabilities vary according to the equipment type and model. Figure 2-2 shows the available IBM LinuxONE 4 models.

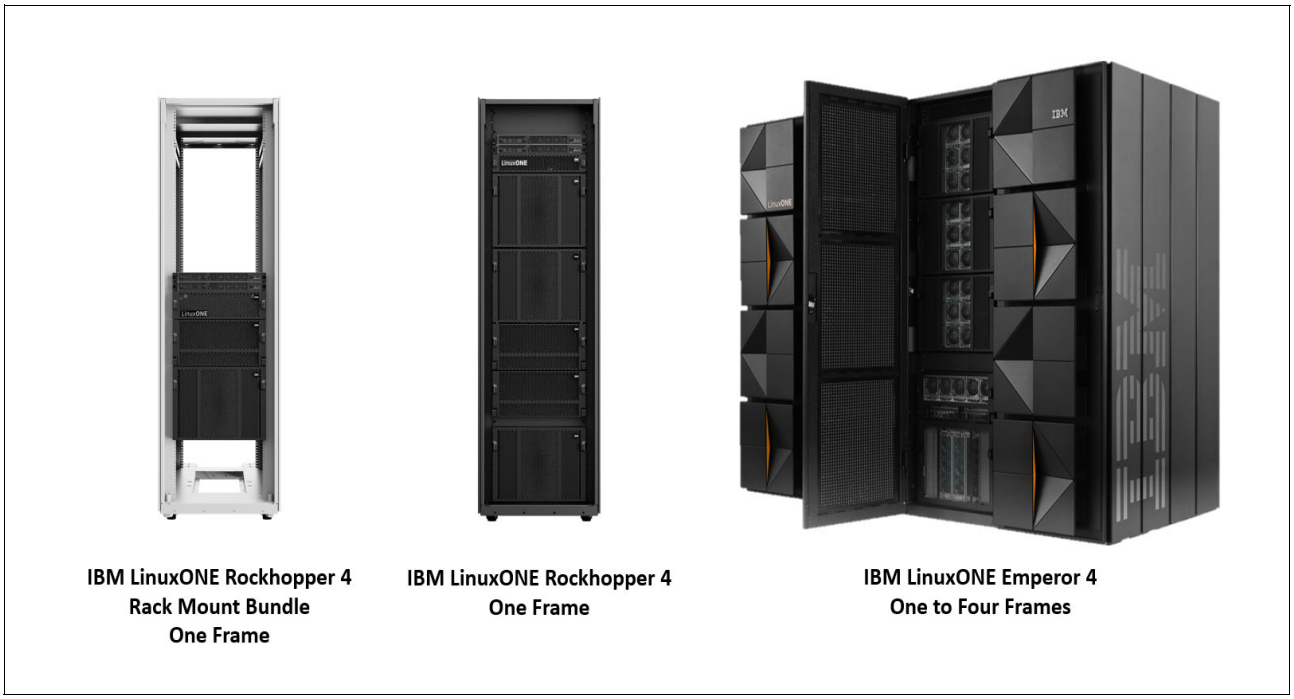


Figure 2-2 IBM LinuxONE 4 available models

LinuxONE reliability, availability, and serviceability

One of the most basic requirements for continuity of service is that the components (hardware and software) in the IT infrastructure are operating in a resilient environment. If you want to attain the highest levels of availability for your business-critical applications and data, you must start with sound fundamentals. These fundamentals include helping ensure that critical IT components have available backup (redundant) capacity, redundant power sources, and redundant connections across critical paths to storage, networks, and other systems, and multiple instances of software (operating systems (OSs), middleware, and applications).

Note: Redundancy by itself does not necessarily provide higher availability. Design and implement your IT infrastructure by using technologies such as system automation and specific features. These technologies can leverage redundancy and respond to failures with minimal impact on application availability.

From a redundancy and resiliency perspective, the IBM LinuxONE platform design (hardware and software) includes RAS principles that are driven by a set of high-level program objectives that move toward continuous reliable operation (CRO) at the system level. The key objectives of IBM LinuxONE are to help ensure data integrity and computational integrity; reduce or eliminate planned and unplanned outages; and reduce the number of repair actions.

The RAS strategy is to manage change by learning from previous generations of IBM Z and LinuxONE and investing in new RAS functions to eliminate or minimize all sources of outages. The RAS strategy employs a building-block approach that is designed to meet stringent requirements for achieving CRO. Here are the RAS building blocks:

- Error prevention
- Error detection
- Recovery
- Problem determination
- Service structure
- Change management
- Measurement
- Analysis

Infrastructure layer

Enhancements to IBM LinuxONE RAS designs are implemented in later IBM LinuxONE platforms by introducing new technology, structures, and requirements. Continuous improvements in RAS are associated with new features and functions to help ensure that the IBM LinuxONE platforms deliver exceptional resiliency. IBM LinuxONE RAS is accomplished with concurrent replace, repair, and upgrade functions for PUs, memory, CPC, and I/O drawers, and I/O features for storage, network, and clustering connectivity. The IBM LinuxONE hardware and firmware are a physical implementation of z/Architecture. Here are the key capabilities that are featured in the infrastructure layer:

- Compute (CPU, memory, and firmware)
- Storage
- Network

Compute

The IBM LinuxONE hardware platform can have one or more frames that are based on the model. The LinuxONE Rockhopper 4 models are always one frame, and the LinuxONE Emperor 4 models can be configured with 1 - 4 frames. Even the model with only one frame has many redundant components and features because resiliency is a design point of these machines. The frames contain the following components:

- CPC drawers, which contain PUs (cores or general-purpose processors), memory, and connectivity to I/O drawers
- I/O drawers with special cores for I/O features
- Special purpose features, such as the on-chip crypto feature and on-chip AI accelerator
- Cooling units for either air or water cooling
- Power supplies
- Oscillator cards (OSCs) for the system clock

The LinuxONE model that you choose depends on the number of PUs, the amount of memory, and how much I/O bandwidth that you require to run your workloads. The PUs, memory, and I/O features have built-in resiliency, and the power, cooling, and system clocks are redundant components in each system.

LinuxONE CPC

To help ensure high resiliency, the IBM LinuxONE CPC contains redundancies for almost all components that are installed to help avoid any type of outage. It also uses redundancy for self-healing. Each CPC drawer in a LinuxONE hosts the PUs, memory, and I/O interconnects. The CPC drawer design aims to reduce, or sometimes eliminate, planned and unplanned outages. The design does so by offering concurrent repair, replace, and upgrade functions for the CPC drawer.

The process that a CPC drawer uses to take over for a failed CPC drawer is called Enhanced Drawer Availability (EDA). EDA enables a single CPC drawer in a multi-drawer configuration to be removed and reinstalled concurrently for an upgrade or a repair.

Figure 2-3 shows a front and back views of a LinuxONE Rockhopper 4 with its covers removed.

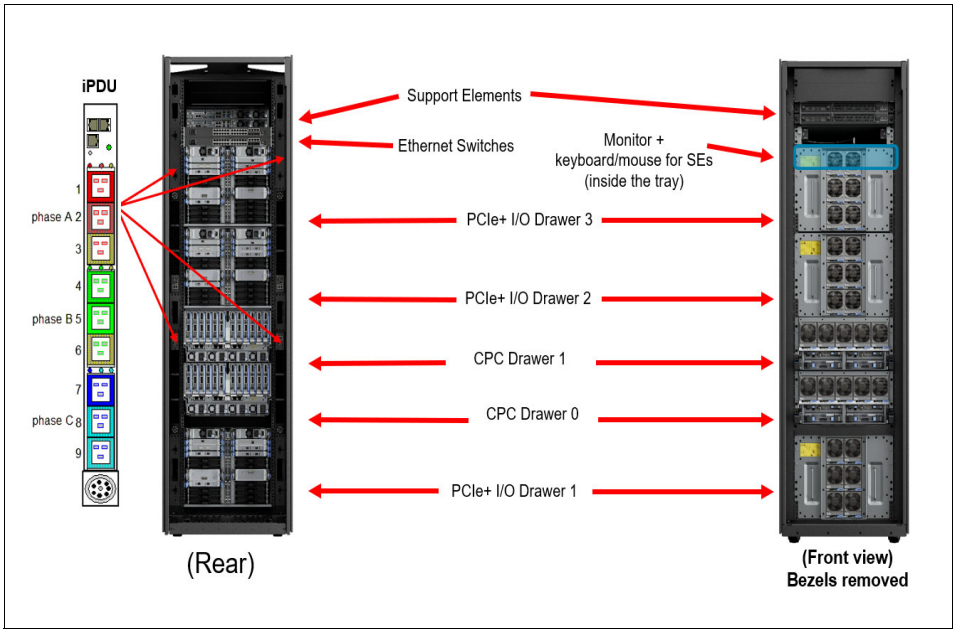


Figure 2-3 BM z16 LinuxONE Rockhopper 4 without its covers (front and rear views)

Processing units

All cores in the LinuxONE platform are physically the same, but are differentiated by their characteristics. The cores can be characterized in advance or dynamically. Certain core characterizations are suited to specific types of tasks or as accelerators. The core characterization types in LinuxONE are as follows:

- An Integrated Facility for Linux (IFL) core/processor is used for Linux and for running the IBM z/VM hypervisor or KVM in support of Linux. z/VM or KVMs often host multiple hundreds of Linux virtual machines (VMs) (known as guests).
- System assist processor (SAP) is used to offload and manage I/O operations. Several SAPs are standard with the platform. More SAPs can be configured if increased I/O processing capacity is needed.

- Integrated Firmware Processor (IFP): Two cores are dedicated to supporting native PCIe features (for example, Remote Direct Memory Access over Converged Ethernet (RoCE) Express, zEDC Express, zHyperLink Express, and Coupling Express), and other firmware functions.
- Central processor (CP): In LinuxONE, a CP may be exclusively used by a GDPS Virtual Appliance, which is a fully integrated software solution to provide continuous availability and disaster recovery (CADR) protection and failover automation of workloads on LinuxONE. For more information, see “GDPS Virtual Appliance” on page 53.

In the unlikely event of a permanent core failure, each core can be individually replaced by one of the available spares. Core sparing is transparent to the OS and applications. The resiliency capabilities for the PUs include the following ones:

- Transparent core sparing
- Concurrent processor drawer repair/add, which includes processor/cache chips, memory, and other internal components
- Transparent SAP sparing

System clocking

LinuxONE has two OSCs for system clocking purposes: one primary and one secondary. If the primary OSC fails, the secondary detects the failure, takes over transparently, and continues to provide the clock signal to the system.

Power

The resiliency capabilities for power for LinuxONE include transparent failover and concurrent repair of all power parts and redundant AC inputs. The power supplies for LinuxONE are also based on the N+1 design. The additional power supply can maintain operations and avoid an unplanned outage of the system.

Cooling

LinuxONE provides an N+1 cooling function for the radiator-based and air-cooled model, which suits the needs of typical business computing. The N+1 (redundant) cooling function for the fluid-cooled model suits the needs of enterprise computing. The resiliency capabilities for cooling include transparent failover and concurrent repair of cooling pumps, blowers, fans, and other items. The single-frame models do not have radiators. The cooling is accomplished by air that is forced by redundant fans.

System control structure

The system control structure includes redundant sideband control access to all units in the platform, and redundant network switches. The Support Elements (SEs) are connected to support processors in the CPC drawer, I/O drawers, power supplies, and cooling units, where Hardware Management Appliances (HMAs) run.

The HMA and SE are appliances that provide hardware management for IBM LinuxONE servers. Hardware platform management covers a complex set of configuration, operation, monitoring, and service management tasks, and other services that are essential to the operations of the platform.

The SEs are stand-alone 1U rack-mounted servers and closed systems that run a set of LinuxONE platform management applications. When tasks are performed at the HMA, the commands are routed to the primary SE of the platform. Then, the primary SE issues those commands to the LinuxONE server.

Two rack-mounted SEs (one is the primary and the other is the alternative) are implemented to manage the LinuxONE platform. SEs include N+1 redundant power supplies. Information is mirrored once per day between the primary and the alternative SEs. The Remote Support Facility (RSF) provides communication with the IBM Support network for hardware problem reporting and service.

Memory

LinuxONE platforms implement Redundant Array of Independent Memory (RAIM), which detects and recovers from failures of dynamic random access memory (DRAM), sockets, memory channels, or Dual Inline Memory Module (DIMM). LinuxONE memory includes these resiliency capabilities:

- DIMM-level failure protection that is based on RAIM technology
- Memory channel and bus protection that is based on CRC and RAIM technology
- Concurrent memory repair/add through the concurrent drawer repair process
- Concurrent memory upgrades

IBM LinuxONE firmware

The IBM LinuxONE firmware provides the flexibility to update dynamically the configuration. You can perform the following tasks dynamically:

- Add a logical partition (LPAR).
- Add a logical channel subsystem (LCSS).
- Add a subchannel set (SS).
- Add a logical core to an LPAR.
- Add a cryptographic coprocessor.
- Remove a cryptographic coprocessor.
- Enable I/O connections.
- Swap processor types.
- Add memory.
- Add a physical core.

To help minimize planned outages, the following tasks are also possible:

- Concurrent driver upgrades
- Concurrent and flexible customer-initiated upgrades
- Concurrent firmware (fixes) updates
- Dynamic processing unit (PU) and SAP reassignment

PCIe fanout

The PCIe fanout in the CPC drawer provides redundant paths for data between memory and the I/O drawers, which house the I/O features. The PCIe fanout is hot-pluggable. If a PCIe fanout fails, a redundant I/O interconnect allows a PCIe fanout to be concurrently repaired without loss of access to its associated I/O domains within the I/O drawer.

PCIe I/O drawer

The I/O drawer supports multiple features that are organized in four hardware domains per I/O drawer. Each domain is through an IBM Interconnect® PCIe switch adapter. The two PCIe switch cards provide a backup path for each other through the passive connection in the I/O drawer backplane. During a PCIe fanout or cable failure, all I/O features in the two domains can be driven through a single PCIe switch card.

I/O features are supported in any combination and can be concurrently added and removed. The resiliency capabilities for I/O include the following items:

- Multiple channel path support
- Concurrent repair/add of all features in an I/O drawer
- Concurrent repair/add of I/O drawer
- Concurrent upgrade of any I/O feature type
- Domain fail-over based on Redundant I/O interconnect
- Dynamic activation of I/O configuration changes

Capacity on Demand (temporary upgrades)

Businesses must handle unpredictable market opportunities, customer needs, and external pressure without missing a beat or interrupting existing processes. This means that your IT infrastructure must support changing business objectives. You should have access to the resources that you need when you need them.

This is the basic principle underlying the Capacity on Demand (CoD) offerings for IBM Z and LinuxONE. With the CoD offerings, you can get the resources that you need when you need them.

The CoD offerings provide permanent and temporary upgrades by activating one or more Licensed Internal Code Configuration Control (LICCC) records. These upgrades occur without disruption to the operation of the server. Depending on the type of upgrade, you can order upgrades yourself by using the Customer Initiated Upgrade (CIU) application on IBM Resource Link® or you can call your IBM sales representative to order the upgrades.

Various dynamic CoD capabilities for resiliency and capacity change on demand are available for IBM LinuxONE:

- Nondisruptive Capacity BackUp (CBU)

  With CBU records, you can replace model capacity or specialty engines into a backup server in an unforeseen loss of server capacity because of an emergency.

- On/Off Capacity on Demand (OOCoD)

  With OOCoD records, you can temporarily add model capacity or specialty engines due to seasonal activities, period-end requirements, peaks in workload, or application testing.

- IBM Z Flexible Capacity for Cyber Resiliency

  With Flexible Capacity for Cyber Resiliency records, you can shift production capacity between participating IBM z16™ servers at different sites. This offering is available for IBM Z beginning with IBM z16 servers and LinuxONE 4.

2.2.2 Physical storage attachment capabilities

This section covers the following topics:

- FICON / Fibre Channel Protocol SCSI
- ECKD over FICON
- Linux Fibre Channel Protocol

FICON / Fibre Channel Protocol SCSI

The storage attachment to an IBM LinuxONE system is realized by using a separate dedicated I/O subsystem that uses separate special cores that are dedicated for I/O. The I/O attachments for the storage servers can use FICON cards. FICON cards can be configured to use the Fibre Channel (FC) protocol for traditional direct-access storage device (DASD) disks in Extended Count Key Data (ECKD)[1] format, or FCP protocol for Fibre Channel (FC) SCSI storage devices.

For resiliency, these cards can be coupled, and they act as a unit that can run the entire workload if one card malfunctions. IBM LinuxONE systems use dedicated processors (SAPs) to handle I/O, which are independent from the application cores. They do not require any capacity planning and are not considered in licensing models for the software stack.

ECKD over FICON

Here are some of the I/O processing characteristics when you use ECKD over FICON:

- The LinuxONE firmware handles multiple paths to a disk.
- I/O requests are serialized and queued in the Linux guest per disk.
- The disk blocks sizes are 4 KB.
- High availability (HA) is handled by the infrastructure.
- DASDs models 3390-3 (2.2 GB), 3390-9 (6.6 GB), 3390-27 (19.8 GB), and 3390-54 (39.6 GB) have predefined sizes. DASD model 3390-A, which uses Extended Address Volumes (EAVs), has a flexible size and support up to 1 TB capacity per disk.

Linux Fibre Channel Protocol

The Linux zFCP device driver adds support for FCP-attached SCSI devices to LinuxONE. FCP is an open, standards-based alternative and supplement to existing FICON connections.

Here are the important I/O characteristics of this type of configuration:

- Several I/Os can be issued for a logical unit number (LUN) concurrently (asynchronous I/O).
- Disk blocks are 512 bytes.
- There is no ECKD emulation overhead for fixed-block devices.
- I/O queues occur in the FCP card or in the storage server.
- No disk size restrictions.
- HA and load balancing is provided by Linux multipathing, type failover, multibus, or a z/VM hypervisor that uses the EDEVICE feature.
- Explores N_Port ID Virtualization (NPIV).[2]

For resiliency purposes, you can use the storage servers through a SAN Volume Controller (SVC), which can build a Storage Stretched Cluster with high consistency and resiliency that is based on the storage controller itself. The SVC can handle storage failovers between different storage servers.

[1] ECKD and Count Key Data (CKD) are used interchangeably in this chapter.

[2] NPIV is an FC standard that makes it possible to create multiple virtual ports on a single physical node port (N-Port), with each virtual port appearing as a unique entity to the FC network.

2.2.3 Physical network attachment capabilities

This section covers the following topics:

- IBM Remote Direct Memory Access over Converged Ethernet Express
- Open Systems Adapter
- Standard Ethernet support
- OSA operating modes
- Open System Adapter Support Facility

IBM Remote Direct Memory Access over Converged Ethernet Express

RoCE cards are similar to Open Systems Adapter (OSA) cards. They can share their ports, but can also run different network protocols in addition to the traditional ETH protocol.

RoCE Express cards use a communication mechanism that is called Shared Memory Communication (SMC), which enables two SMC-capable peers to communicate by using a memory buffer that is allocated by each peer for the partner's use. There are two types of SMC:

- Shared Memory Communications over Remote Direct Memory Access (SMC-R)

  SMC-R is a network capability that strengthens performance for sharing data and reducing data transmission network overhead. SMC-R provides application-transparent usage of this new RoCE feature, which reduces the network overhead and latency of data transfers, effectively offering the benefits of optimized network performance between two LinuxONE processors.

- Shared Memory Communications-Direct Memory Access (SMC-D)

  SMC-D enables direct access to shared virtual memory by providing a highly optimized network interconnect for IBM LinuxONE intra-CPC communications by using Internal Shared Memory. For more information, see "Internal Shared Memory".

Internal Shared Memory

Internal Shared Memory (ISM) provides adapter virtualization with high scalability that enables LPARs (or guest VMs) to logically share virtual memory. ISM enables SMC-D, and provides savings in physical resources with performance benefits.

Open Systems Adapter

The OSA is a network controller that you can install in a LinuxONE I/O cage. The OSA-Express features integrate network interface hardware and support many networking transport protocols. Every OSA-Express feature provides capabilities in the following areas:

- Standard Ethernet support
- Operating modes
- Connectivity options (bandwidth and data throughput)
- RAS

Effectively, the OSA integrates the control unit and device into the same hardware. It does so by placing it on a single card that directly connects to the CPC I/O bus.

Standard Ethernet support

The following Ethernet standards are applicable for OSA-Express features (standard transmission schemes):

- IEEE 802.3 and IEEE 802.1
- Ethernet 2.0
- Data Integrity Extensions (DIX) 2

OSA operating modes

With the IBM LinuxONE platform, the integration of a channel path with network ports makes the OSA a unique channel or channel path identifier (CHPID) type, which is recognized by the hardware I/O configuration on a port-by-port basis. The following CHPID types are described in further detail in *IBM Z Connectivity Handbook*, SG24-5444 and in later sections in this book:

- Queued direct input/output (QDIO) (OSD)
- Non-QDIO (OSE)[3]

Open System Adapter Support Facility

Open System Adapter Support Facility (OSA/SF) is a host-based tool that is used to customize and manage OSA-Express features:

- OSA/SF is not required for the OSA feature that is configured for the QDIO mode or the default IP pass-through non-QDIO mode. However, it can be used for problem determination purposes.
- OSA/SF is a required facility when the OSA feature is being configured for shared non-QDIO mode and where System Network Architecture (SNA) definitions are involved.
- With the IBM LinuxONE platform and OSA/SF, the Hardware Management Console (HMC) is enhanced to provide configuration, validation, activation, and display support for the OSA-Express7S, OSA-Express7S 1.2, and OSA-Express6S features:
  - OSA/SF on the HMC is required for OSA-Express7S 1.2, OSA-Express7S, and OSA-Express6S features.
  - One OSA/SF application can communicate with all OSA features in an IBM Z platform.
  - OSA/SF communicates with an OSA feature through a device (type OSD) that is defined by using Hardware Configuration Definition (HCD) or the Input/Output Configuration Program (IOCP).

2.3 Operating environments

This section introduces the IBM LinuxONE platform virtualization capabilities, such as Processor Resources Systems Manager (PR/SM) and DPM. It also describes virtualization implementations with z/VM and KVM hypervisors and container platforms.

2.3.1 Virtualization

IBM LinuxONE is a virtualized environment that accepts ‘bare metal’ as the virtualization mode with the Processor Resource/System Manager (PR/SM) entity. This level is already a virtualization layer that is implemented in the hardware and is highly efficient for resiliency and failover capabilities between LPARs and the resources that are attached to it.

An LPAR is a subset of the processor hardware that is defined to support an OS. An LPAR contains resources (processors, memory, and input/output devices) and operates as an independent system. Multiple LPARs can exist within LinuxONE mainframe hardware system.

[3] IBM LinuxONE 4 is planned to be the last server to support OSE networking channels. This refers to IBM support for the SNA protocol that is transported natively out of the server by using OSA-Express 100BASE-T adapters.

Partitioning control specifications are partly contained in the Input/Output Configuration Data Set (IOCDS) and are partly contained in a system profile. The IOCDS and profile both are in the SE. An IOCDS contains information to define the I/O configuration to the processor channel subsystem and is created by a program called IOCP.

A second layer of virtualization is in each LPAR and can be realized by implementing a hypervisor in the LPAR, which enables the virtualization of all hardware resources, and thousands of Linux guests, which can be run concurrently in a single LinuxONE system.

The two hypervisors that can be installed in a LinuxONE LPAR are z/VM, and KVM:

- With z/VM, you can have the premier IBM hypervisor that has matured over decades and has leveraged granular resource sharing and resource shifting between the virtualized Linux virtual guests that are controlled by z/VM.
- With KVM, you can enable on LinuxONE, which is a hypervisor that is developed by the Open Source Community, LinuxONE was adapted by Red Hat, SUSE, and Canonical.

Platform virtualization

Platform virtualization is a principal strength of IBM LinuxONE. It is embedded in the architecture and built into the hardware, firmware, and OSs. For decades, the IBM LinuxONE platforms have been designed based on the concept of partitioning resources (such as CPU, memory, storage, and network resources). So, each set of features can be used independently with its own operating environment.

Every LinuxONE platform is highly virtualized, with the goal of maximizing utilization of computing resources while lowering the total number of resources and cost that are needed to run critical workloads and solutions.

Virtualization can help secure and isolate application workloads and data within virtual servers and storage devices for replication and restoration. With this added resiliency, you have greater flexibility in maintaining a highly available infrastructure while performing planned maintenance and in configuring low-cost DR solutions.

2.3.2 IBM Processor Resource Systems Manager

The hardware-level virtualization capability in IBM LinuxONE, PR/SM, is a Type-1 hypervisor that runs directly on bare metal so that you can create multiple LPARs. PR/SM is a highly reliable and secure firmware-encapsulated virtualization technology that enables multiple OSs to run on the same physical platform. Each OS runs in its own LPAR.

PR/SM is a technology that is used in IBM LinuxONE to provide logical partitioning and resource management capabilities. PR/SM enables a LinuxONE system to be divided into multiple LPARs, each running its own OS instance and set of applications with the highest isolation.

PR/SM logically partitions the platform across the various LPARs to share resources, such as processor units and I/O (for networks and storage), which enables a high degree of virtualization and the highest isolation concurrently.

The main goal of PR/SM is to maximize the usage of system resources and improve overall system performance. It enables up to 85 LPARs to run concurrently on a single physical LinuxONE Emperor server and up to 40 LPARs on a single LinuxONE Rockhopper, enabling organizations to consolidate workloads while isolating them for resiliency and reducing hardware costs.

Here are some key features and capabilities of IBM PR/SM:

- Logical partitioning: PR/SM divides a physical LinuxONE into LPARs. Each LPAR functions as an independent virtual LinuxONE with its own processors, memory, and I/O devices. This enables different OSs to run concurrently on the same hardware.
- Resource management: PR/SM manages the allocation of system resources, such as processors, memory, and I/O channels to each LPAR based on predefined rules and policies. It helps ensure that each partition receives the necessary resources to meet its workload requirements while preventing resource contention and optimizing overall system performance.
- Dynamic resource allocation: PR/SM supports dynamic resource allocation, enabling system administrators to adjust the allocation of processors and memory to individual partitions without disrupting the operation of other partitions. This flexibility enables organizations to respond to changing workload demands and optimize resource utilization dynamically.
- HA and failover: PR/SM provides built-in resiliency features for high availability and disaster recovery (HADR). It supports the usage of redundant components, such as processors and I/O devices to help ensure system availability during hardware failures. Also, PR/SM permits automation to support workload movement or restart on different physical machines for planned maintenance or in failures, which minimizes downtime.
- Workload isolation: Each LPAR in PR/SM is isolated from the others, which provides a secure and controlled environment for running different workloads. This isolation has Evaluation Assurance Level (EAL) 5+ Common Criteria certification, which prevents disruptions that are caused by one partition from affecting others and improves system stability and security.
- LPAR weights: In an environment with shared capacity, use LPAR weight capabilities to avoid capacity stealing between LPARs. A definition of capacity helps ensure certain workloads in the LPARs. For more information, see z/VM Education: What's going wrong? LPAR Weights and 2.8.3, “Automation with Linux HA components” on page 48.

Overall, PR/SM is a powerful virtualization technology that enables efficient resource management and workload consolidation on IBM LinuxONE systems. It helps organizations maximize the usage of their LinuxONE hardware and achieve better performance, flexibility, and reliability in their computing environments.

Figure 2-4 shows the various options for leveraging the IBM PR/SM virtualization capabilities, such as z/VM, KVM, and native z/OS and native Linux distributions. IBM LinuxONE supports only z/VM, KVM, and native Linux LPARs.

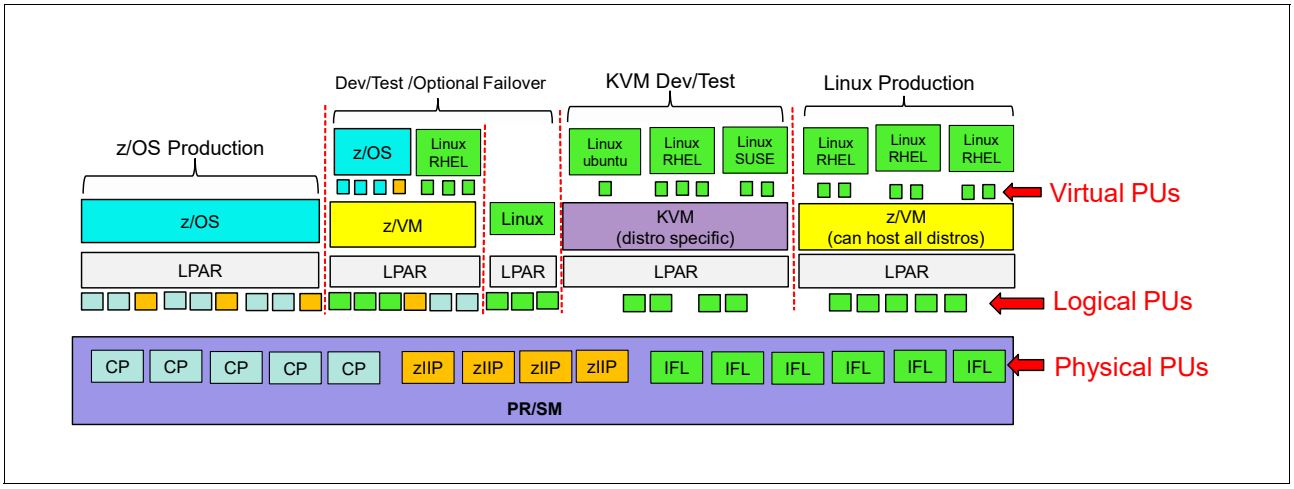


Figure 2-4 Leveraging PR/SM virtualization capabilities

Resiliency in workloads needs the usage of one or more LPARs along with various modes of software-level control to help ensure availability. Predetermine the availability levels for each workload to understand the corresponding requirements.

To provide application fail-over capabilities for multi-LPAR application workloads, implement HA clustering solutions such as Pacemaker, Red Hat High Availability Add-on for Red Hat Enterprise Linux, or SUSE HA. These solutions facilitate automatic failover and workload recovery if there are hardware or software failures.

Regular testing of LPAR application workloads' resiliency through simulation and controlled failure scenarios is vital. Conducting DR drills, fail-over testing, and load testing validate the effectiveness of resilience measures and identifies areas for improvement.

Developing a comprehensive DR plan that is tailored to your specific needs is critical. This plan should encompass backup and replication strategies, data replication, fail-over procedures, and data integrity validation, which are all aimed at expediting recovery processes and minimizing downtime.

LinuxONE can be configured in either PR/SM or DPM mode.

2.3.3 Dynamic Partition Manager

DPM is a configuration manager that is designed for setting up and managing Linux servers that run on a LinuxONE system. On a DPM-enabled system, the runtime environment for your LinuxONE is called a *partition*, which is also the runtime environment for a hypervisor and its guest OS images.

On a LinuxONE system, a partition is a virtual representation of all physical hardware resources of that system, which include processors, memory, and input/output (I/O) adapters. On LinuxONE systems, as on other platforms, an adapter is a physical device that connects the system to other computers or devices. In contrast to other platforms, adapters on a LinuxONE system can be shared between partitions, which reduces the number of adapters that might be required to handle a specific workload.

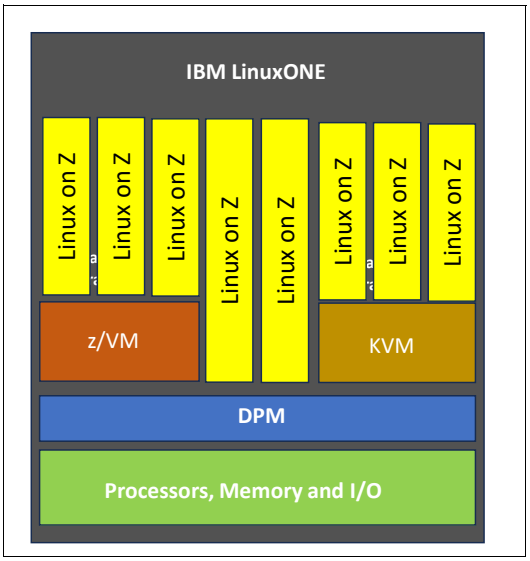


Figure 2-5 Dynamic Partition Manager diagram

DPM is a feature that is enabled by the systems' initialization process and is mutually exclusive with PR/SM. When it is enabled, the DPM mode is ready when the system starts.

With DPM, use the HMC to configure the running environment for your LinuxONE server. DPM automatically discovers and displays the system resources that are available for you to use and indicates how your selections might affect other servers and applications that are already defined or running on the same system.

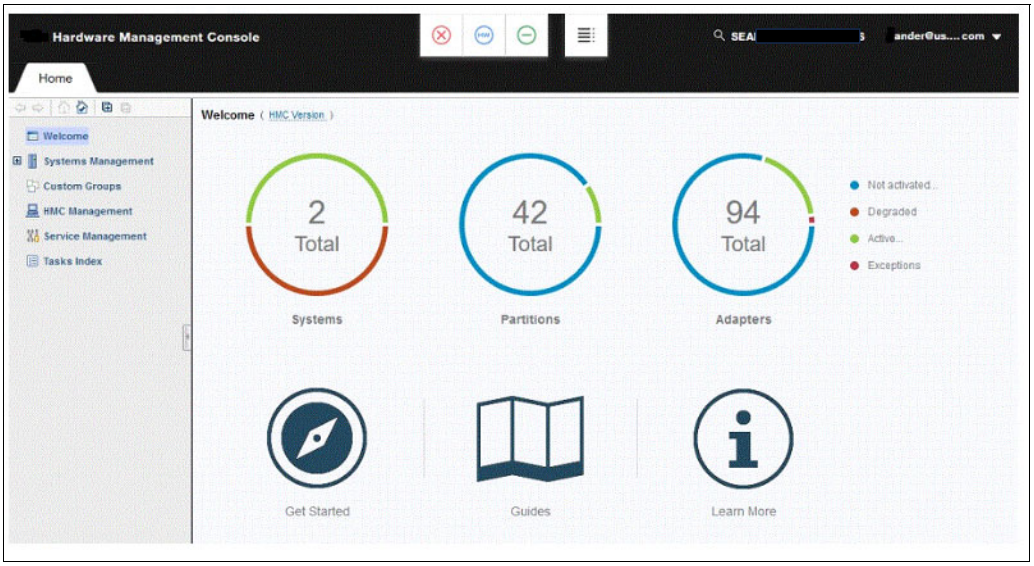


Figure 2-6 HMC "Welcome" page in DPM mode

After your system is running in DPM mode, you can use DPM to complete the following tasks:

- Modify system resources without disrupting running workloads.
- Monitor sources of system failure incidents and conditions or events that might lead to workload degradation.
- Create alarms so that you can be notified of specific events, conditions, and changes to the state of system resources.
- Update individual partition resources to adjust capacity, redundancy, availability, or isolation.

For more information about DPM mode, see *Dynamic Partition Manager (DPM) Guide*, SB10-7176.

2.3.4 Highest workload isolation in LPAR

The LPARs in an IBM LinuxONE system have the highest isolation (certified as Evaluation Assurance Level 5 (EAL5), which is the highest isolation in the IT industry. Therefore, workloads in LPARs have two unique resiliency characteristics on IBM LinuxONE: The resources can be shared across LPARs and reach the highest workload isolation and resiliency in an LPAR concurrently.

Two LPARs can be used as separate entities with shared computing resources. If one LPAR is not functioning, the other LPAR can run the entire workload due to the capacity shift to it. This is built-in resiliency per hardware design.

As shown in Figure 2-5 on page 29, DPM does not cover all features of PR/SM, but it enhances the user experience for an IBM LinuxONE, especially in a cloud-based environment where software-defined hardware is important and hardware resiliency is controlled and managed with external tools.

2.3.5 The z/VM hypervisor on IBM LinuxONE

IBM z/VM is a highly secure and scalable virtualization technology. It is a hypervisor that runs on LinuxONE systems, enabling multiple VMs to run on a single or multiple LPARs on a physical machine. Each VM can run its own OS, such as Linux. With z/VM, organizations can consolidate workloads and improve resource utilization and resiliency while maintaining high levels of security and performance.

z/VM is a Type-2 hypervisor that enables sharing of the LinuxONE platform's physical resources, such as disk, memory, network adapters, and cores (called IFL). These resources are managed by the z/VM hypervisor, which runs in an LPAR and manages the VMs that run under the control of the z/VM hypervisor. Typically, the z/VM hypervisor is used to run Linux virtual servers, but it can also be nested, which means a second layer virtualization is possible. This enables granular testing capabilities and verification for resiliency.

With z/VM, the workload can scale efficiently horizontally by enabling more virtual servers in the same z/VM and vertically, which means that applications can scale without modification by dynamically dedicating more resources to them through the z/VM hypervisor.

You can combine the horizontal and vertical scalability and reach the best virtualization capability in an IT environment, which is unique to the IBM LinuxONE environment.

z/VM can host Linux guests from all distribution partners in the same LPAR because its EAL4+ certified isolation for virtual server helps ensure the isolation of each server in a safe, secure, and scalable environment.

Within z/VM, the network topology can be simplified by using z/VM Virtual Switch (VSWITCH) technology, which is a virtual, software-defined network that can run in Layer 2 or Layer 3 and connects to one or multiple OSA network cards for resiliency.

A z/VM VSWITCH that is connected to multiple OSA cards can implement port aggregation, which enables resilient failover and enlarged network bandwidth concurrently.

The direct connection to a LinuxONE inter-network topology that is implemented with HiperSockets across LPARs is another unique characteristic that uses the resiliency of the hardware features that are built into LinuxONE and the z/VM hypervisor.

z/VM resiliency

z/VM delivers resiliency for the VMs and workloads at several levels:

- Linux multipathing of ECKD and Fixed-Block Architecture (FBA) emulated devices (EDEVs or EDEVICEs). For ECKD devices, this feature is called HyperPAV.
- A capacity for attaching NPIV FCP devices directly to the guests for Linux multipathing.
- Network redundancy with 2 or more OSAs through VSWITCH.

  The z/VM VSWITCH supports various configurations, which include Layer 2 and 3 networking, Active/Backup, IEEE 802.3ad Link Aggregation, Virtual Ethernet Port Aggregator (VEPA) IEEE 802.1Qbg, and VLAN aware and unaware. Other than OSA Express, RoCE Express does not support z/VM VSWITCH technology to provide path redundancy.
- z/VM Single System Image (SSI) and Live Guest Relocation (LGR):
  - SSI is a concept in z/VM that enables up to eight z/VM instances to function as a single system. With SSI, these instances are interconnected and collectively manage a shared set of resources, such as storage and network infrastructure. The goal is to provide a cohesive and transparent environment for running VMs across the interconnected z/VM instances.
  - LAN is a feature of z/VM SSI that enables the movement of running VMs from one z/VM instance to another one while the VMs continue to operate without interruption. These features provide flexibility and HA by enabling workload balancing, DR, and maintenance operations without affecting the availability of applications running in the VMs

For effective workload balancing, prioritizing virtual guests in the LPAR is helpful. Therefore, you should help ensure that certain VMs always have enough capacity and get as much capacity that is available if possible. This technology of VM SHARE can be defined on the VM level. For more information, see SET SHARE Command.

Another concept for effective workload definition is CPU pooling with z/VM, which uses a pool of CPU capacity within an LPAR and distributes it on a VM base to the workloads. This is used for effective licensing within an LPAR. For more information, see Using CPU Pools.

z/VM Single System Image

A z/VM SSI cluster is a multisystem environment in which the z/VM systems can be managed as a single resource pool and guests can be moved from one system to another one while they are running. SSI is included as part of the z/VM base.

A z/VM SSI cluster consists of up to eight z/VM systems in an Inter-System Facility for Communications (ISFC) collection. ISFC is a function of the z/VM Control Program (CP) that provides communication services between transaction programs on interconnected z/VM systems. A group of interconnected domains consisting of z/VM systems that use ISFC to communicate with each other is known as an ISFC collection.

Each z/VM system is a member of the SSI cluster. Figure 2-7 shows the basic structure of a cluster with four members. The cluster is self-managed by CP by using ISFC messages that flow across channel-to-channel (CTC) devices between the members. All members can access shared DASD volumes, the same Ethernet local area network (LAN) segments, and the same storage area networks (SANs).

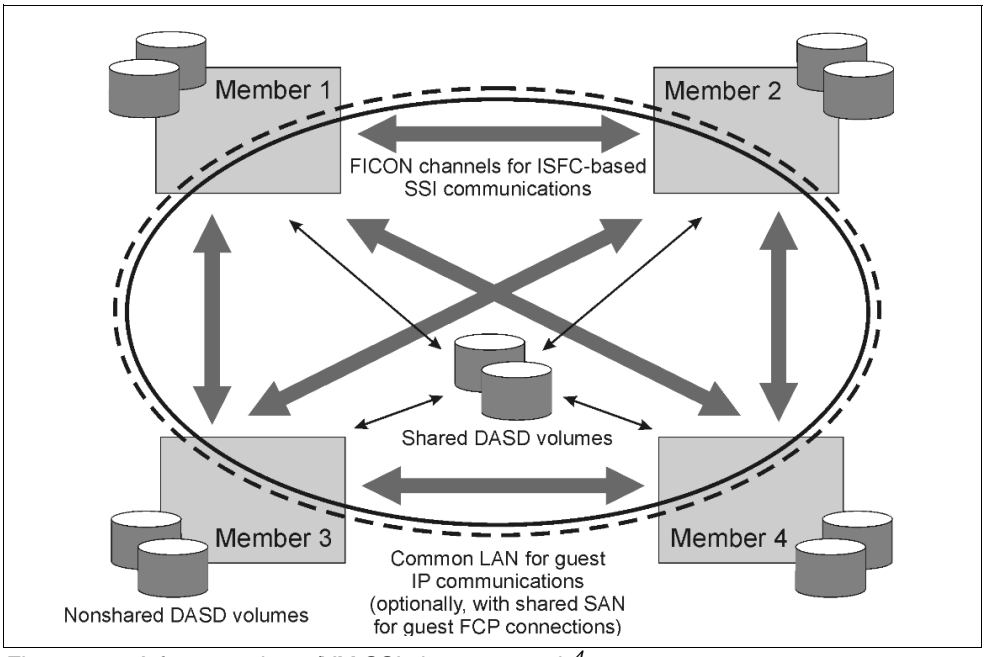


Figure 2-7 A four-member z/VM SSI cluster example[4]

SSI enhances the z/VM systems management, communications, disk management, device mapping, VM definition management, installation, and service functions to enable multiple z/VM systems to share and coordinate resources within an SSI structure. This combination of enhanced functions provides the foundation that enables LAN, which is the ability of a Linux guest to move from one z/VM system to another one within the SSI cluster.

z/VM Live Guest Relocation

A z/VM SSI cluster provides a virtual server mobility function that is called an LGR.

A running virtual server (guest VM) can be relocated from one member to another one. Relocating virtual servers can be useful for load balancing and 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 so that you can maintain z/VM and keep your Linux virtual servers highly available. When a relocation is initiated, the guest continues to run on the source member until the destination environment is prepared. Then, the guest briefly quiesces on the source member and then resumes on the destination member. For more information, see Planning for a Single System Image and Live Guest Relocations.

[4] A z/VM SSI cluster can have up to eight members.

2.3.6 The KVM hypervisor on IBM LinuxONE

KVM is a hypervisor that is developed by the Open Source community. It is adapted and integrated in every Linux distribution for IBM LinuxONE.

KVM represents a hypervisor that provides server virtualization for Linux workloads that run on the IBM LinuxONE platform, which leverages the existing skill set that is used in other hardware architectures. It is also a Type-2 hypervisor that runs in an LPAR and enables sharing CPUs (IFLs), memory, and I/O resources through platform virtualization.

KVM enables the creation and running of multiple VMs on a Linux host, which provides full hardware virtualization capabilities. It leverages the hardware virtualization extensions that are available on IBM LinuxONE systems for efficient and secure virtualization.

It can coexist with z/VM environments and is optimized for scalability, performance, security, and resiliency. It provides standard Linux and KVM interfaces for simplified virtualization and operational control.

The KVM hypervisor is integrated into these major supported Linux distributions:

- Red Hat Enterprise Linux Server
- SUSE Linux Enterprise Server
- Canonical Ubuntu

KVM on LinuxONE relies on hardware, LPAR, and Linux host redundancy and resiliency as a starting point. Therefore, it implements features of HA, redundancy, and resiliency at the hypervisor level, such as the following ones:

- Multipathing of virtual disks with Linux Device Mapper-Multipathing (DM-MP).
- Network redundancy that is based on Linux bonding and teaming. Active-backup, round-robin, and IEEE 802.3ad Link Aggregation are supported.
- Virtual Machine Live Migration, which enables you to move a running VM from one physical host to another one without disrupting its operation.
- Live Migration: The process of transferring a running VM from the source host to the destination host while it continues to run without any noticeable downtime. Use it to achieve load balancing, perform hardware maintenance, or migrate VMs between different physical hosts for various reasons, such as a workload balancing or a planned maintenance.
- Because RoCE Express does not provide a promiscuous mode, you cannot use Open VSWITCH in a KVM host to provide path redundancy for its guests; instead, use MacVtap for internal communication of KVM guests.

KVM Cluster

KVM Cluster is a virtualization environment that leverages the KVM hypervisor to create and manage VMs on Linux systems. KVM Clusters connect multiple nodes to work collectively and manage virtual resources. KVM Clusters implement HA and continuous operations (CO) by distributing VMs across nodes. If there is a node failure, the VM can migrate to operational nodes, which minimizes downtimes.

2.3.7 Container platforms

IBM LinuxONE offers robust support for various container technologies and platforms, which enables the usage of cloud-native applications like Linux-based workloads, microservices, hybrid cloud deployments, and their associated ecosystems. The following configuration options are available:

- Linux applications
- Microservices architecture
- Cloud-native applications
- Hybrid cloud deployments
- Container platforms
- Parallel Access Volume
- HyperPAV
- ECKD with HyperPAV
- Using system internal network capabilities

Linux applications

IBM LinuxONE facilitates the containerization of Linux-based applications through technologies such as Open Container Initiative (OCI) compliant containers. Containers provide a lightweight and isolated runtime environment, which helps ensure consistency and portability across different environments. The LinuxONE powerful hardware resources and scalability make it an optimal choice for running large-scale containerized workloads.

Microservices architecture

LinuxONE is highly suitable for deploying microservices-based architectures by using containers. By breaking down complex applications into smaller, decoupled services, developers can independently develop, deploy, and scale these services. Containers enable efficient resource utilization and rapid deployment, which makes LinuxONE an ideal platform for managing microservices workloads.

Cloud-native applications

IBM LinuxONE fully supports the development and deployment patterns of cloud-native applications. These applications are designed to harness the benefits of containerization, scalability, and elasticity. Leveraging technologies such as Kubernetes, IBM LinuxONE enables automated container orchestration, scaling, and load balancing, which provide a reliable and high-performance foundation for running cloud-native applications.

Hybrid cloud deployments

IBM LinuxONE facilitates the deployment of containerized workloads in hybrid cloud environments. Using solutions like Red Hat OpenShift Container Platform, which supports multiple hardware architectures like IBM LinuxONE, enables the management and deployment of containers across both on-premises LinuxONE systems and public cloud platforms. This approach enables users to leverage containerization advantages while maintaining a consistent and secure environment throughout their hybrid cloud infrastructure.

Container platforms

LinuxONE supports many container platforms that provide capabilities for managing containerized applications. An example is the Red Hat OpenShift Container Platform, which is available for various hardware architectures, which include IBM LinuxONE. Red Hat OpenShift Container Platform, built on Kubernetes, offers a comprehensive environment for developing, deploying, and managing containerized applications. It includes features like automated scaling, load balancing, and container lifecycle management.

2.4 LinuxONE virtual storage

This section covers the virtual storage capabilities that are leveraged in the LinuxONE platform. The topics include Parallel Access Volumes (PAVs), HyperPAV, HyperSwap capabilities, and multipathing.

The PAV facility enables a controller to offer multiple device numbers that resolve to the same DASD, which enables I/O to the same DASD concurrently.

The device that is addressed by the OS to perform the I/O operation is the “base” device. When the base device is busy working and another I/O operation to the same base address starts, it can be run by one of the “alias” devices, if PAV or HyperPAV is used.

If there is no aliasing of disk devices, then only one I/O transfer can be in progress to a device at a time regardless of the actual capability of the storage server to handle concurrent access to devices. PAV exists for Linux on IBM Z in the form of PAV and HyperPAV. Compared to PAV, HyperPAV is simpler to administer and provides greater flexibility. PAV and HyperPAV are optional features that are available on the IBM DS8000® Storage System series.

Parallel Access Volume

PAV uses “base” and “aliases” devices. Base devices are the target device to carry the I/O operation. With PAV, if a base device is busy and an alias is available, it is used to perform the I/O.

Figure 2-8 shows a PAV example.

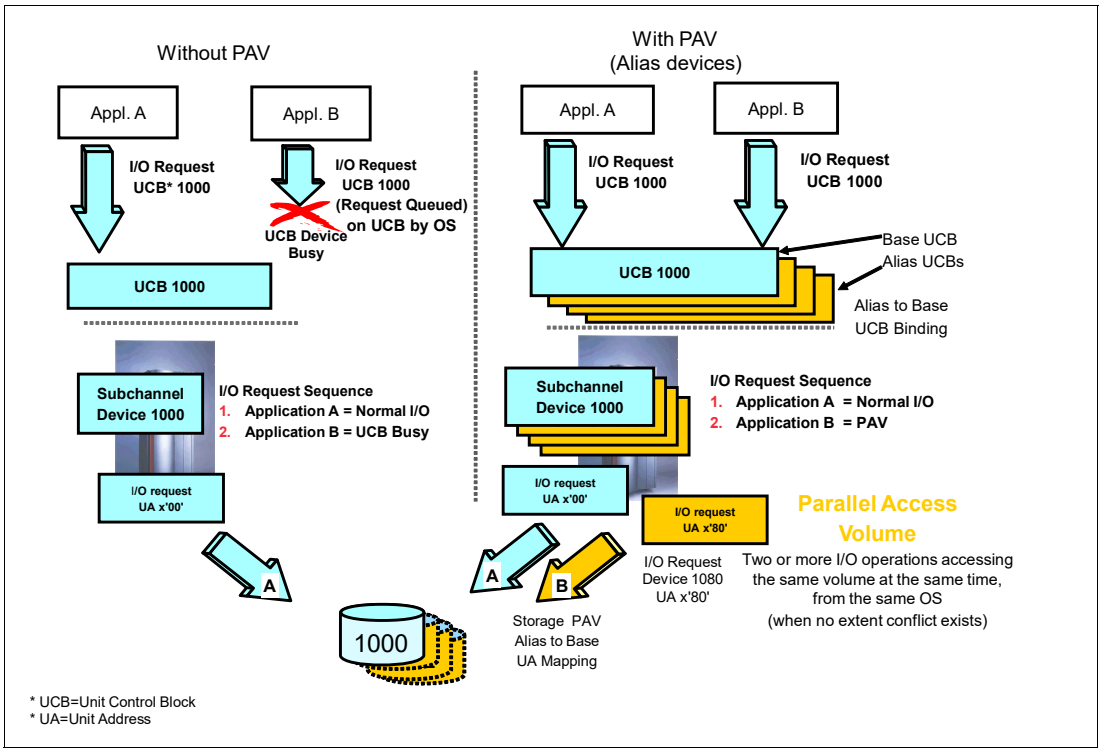


Figure 2-8 Parallel Access Volume example

HyperPAV

The PAV capability is extended dynamically with IBM Hyper-Parallel Access Volume (HyperPAV). HyperPAV enables multiple I/O operations to a DASD through a pool of base and alias subchannels (devices) that are shared within a logical subsystem (LSS). HyperPAV eliminates the need for users to map volumes to aliases and takes care of the aliases and I/Os automatically.

ECKD with HyperPAV

The base and HyperPAV devices (alias) are defined on the storage server and on the IOCDS on LinuxONE, where the DASD devices are defined with unit addresses. A Logical Control Unit (LCU) is a logical set of DASD devices on a DS800 series disk storage unit, and it can have up to 256 possible DASD device addresses (00 - FF). After the base devices are defined, any remaining device numbers in the LCU can be used as an alias by the system to access any of the base addresses in the same LCU.

Using ECKD devices with HyperPAV (Figure 2-9) can improve performance and increase availability. Here are some characteristics of ECKD with HyperPAV:

- The DASD driver sees the real disk and all aliases.
- Load-balancing with HyperPAV is done in the DASD driver.
- It uses fewer processor cycles than Linux multipathing.

For resiliency, a storage server with ECKD disks typically is mirrored through the storage server function of Metro Mirror (MM) for synchronous replication and metro distances, or with Global Mirror (GM) if the distances are larger. The swapping between the mirrored disks must be triggered and managed by external tools, such as GDPS. For more information about HyperPAV support, see LinuxONE and Linux on IBM Z HyperPAV.

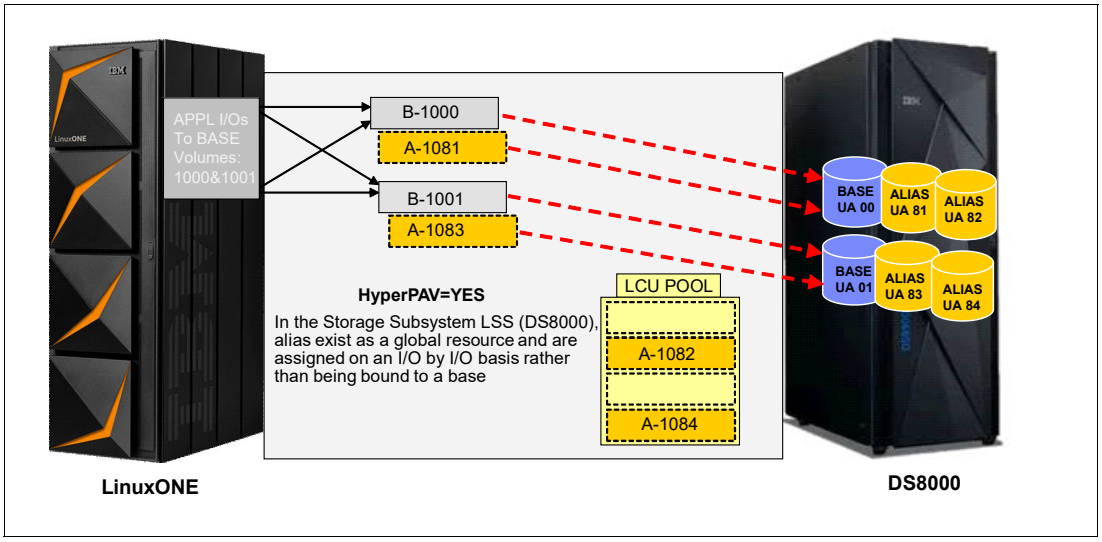


Figure 2-9 HyperPAV exploitation example

2.5 LinuxONE Virtual Network

The IBM LinuxONE system is a fully virtualized platform that can support many LPARs and thousands of Linux system images concurrently. Therefore, network connectivity covers the connections between the platform and external networks and the intra-system connectivity between the Linux system images or VMs that are hosted on a system.

Each LPAR can have one or more network interfaces that are assigned to it. The network topology depends on the isolation requirements and the characteristics of the workloads, but sharing and isolation are concepts that are widely used in an IBM LinuxONE environment. The network cards can have multiple ports that represent an individual network interface. These ports can be assigned to one or multiple LPARs, so the isolation of the LPAR and the sharing of a port are unique characteristics of IBM LinuxONE and contribute to the high resiliency of the environments that are hosted there.

An IBM LinuxONE system can use multiple network cards and different network topologies.

Using system internal network capabilities

With IBM LinuxONE, there is a possible network topology for internal networks among LPARs that uses microcode only and does not require a network card.

2.5.1 IBM HiperSockets

IBM HiperSockets is an integrated function of LinuxONE that supplies attachments to virtual LANs within the same system. HiperSockets provides connectivity across multiple LPARs on the same LinuxONE system by performing memory-to-memory data transfers through microcode in the LinuxONE hardware in a secure way. The HiperSockets function eliminates the usage of I/O subsystem operations and external network connections.

For more information about the available OSA Express and RoCE Express features, see *IBM Z Connectivity Handbook*, SG24-5444.

Depending on the hypervisor, you can use virtual switches with the network capabilities of IBM LinuxONE.

Figure 2-10 shows examples of the most used network capabilities with OSA, RoCE, and HiperSockets network interfaces. There are three Linux instances: Two of them run as z/VM guests in one LPAR, and a third Linux instance runs in another LPAR. Within z/VM, Linux instances can be connected through a guest LAN or VSWITCH. Within and among LPARs, you can connect Linux instances through HiperSockets. OSA-Express cards that run in either non-QDIO mode or in QDIO mode can connect the LinuxONE to an external network.

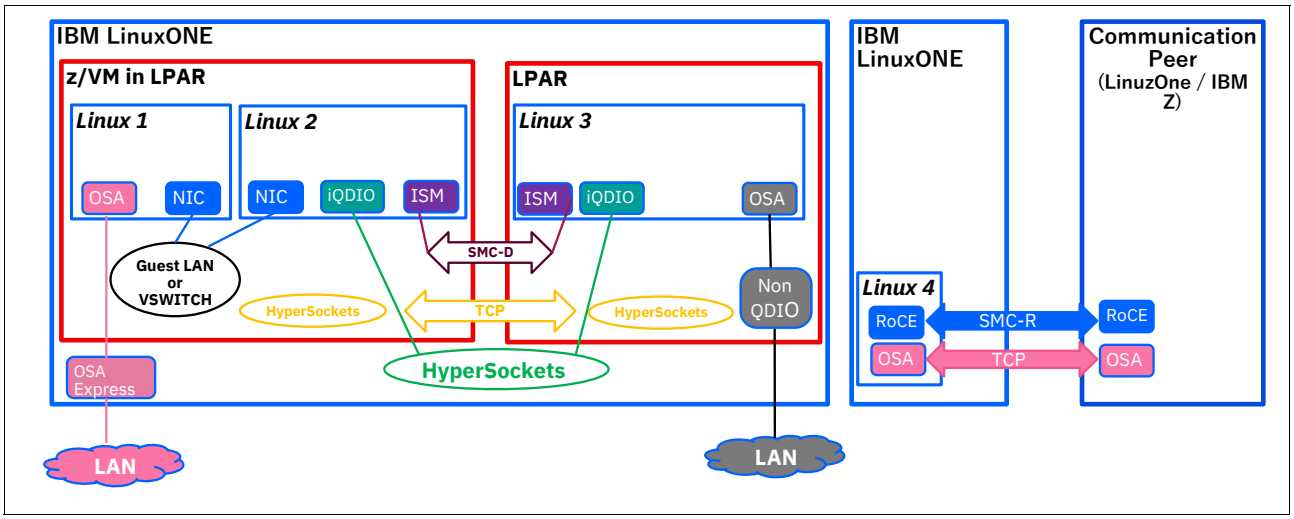


Figure 2-10 LinuxONE network capabilities

Note: For more information about QDIO and Non-QDIO modes, see *IBM Z Connectivity Handbook*, SG24-5444.

For resiliency, it is a best practice to use multiple network interfaces, even for different workloads. It can be beneficial to use multiple networks with different characteristics to support the application requirements regarding network bandwidth and resiliency.

2.5.2 KVM MacVTap

MacVTap is a Linux kernel device driver that enables bridge-like networking and simplifies the virtualized, bridged networking setup. The MacVTap driver attaches the guest's network interface card (NIC) directly to a specified physical interface on the host machine. It makes both the guest and the host appear directly on the network switch that the host is connected to. MacVTap also improves throughput and latencies to external systems.

A MacVTap endpoint is a character device that follows the **`tun/tap ioctl`** interface and can be used directly by kvm/qemu and other hypervisors that support the **`tun/tap`** interface.

MacVTap can be configured in any of three different modes, which determine how the MacVTtap device communicates with the lower device in the KVM host. The three possible modes are VEPA, Bridge, and Private mode. For more information, see Configuring a MacVTap interface.

VEPA

VEPA is the default mode. Data flows from one endpoint through the source device in the KVM host to the external switch. If the switch supports hairpin mode, the data is sent back to the source device in the KVM host and then sent to the destination endpoint.

Bridge

Bridge mode connects all endpoints directly to each other. Two endpoints that are both in bridge mode can exchange frames directly without a round trip through the external bridge. This is the most useful mode for setups with classic switches, and when inter-guest communication is performance critical.

Private

Private mode behaves like a VEPA mode endpoint in the absence of a hairpin aware switch. Even when the switch is in hairpin mode, a private endpoint can never communicate to any other endpoint on the same lowerdev.

2.5.3 z/VM Virtual Switch

VSWITCH is a bridge-like network that emulates switch technology inside the hypervisor layer. The virtual switch is designed to help eliminate the need for VMs that act as routers. Virtual routers consume valuable processor cycles to process incoming and outgoing packets, which require extra copying of the data that is transported. The VSWITCH helps alleviate this problem by moving the data directly between the real network adapter and the target or originating guest data buffers.

VSWITCH is a z/VM 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 the appropriate QDIO NIC and connect it to this switch by using the **`NICDEF`** directory statement. Alternatively, a z/VM user can create a virtual network adapter (with the CP **`DEFINE NIC`** command) and connect it to this LAN (with the **`COUPLE`** command). For more information about these commands, see z/VM: CP Commands and Utilities Reference.

The VSWITCH can be configured in Bridge or VEPA mode, depending on your requirements for network separation, which enables either a performance boost of the “hairpin turn” or a separation out to the physical switch. Also, the virtual switch supports link aggregation, which is a combination of multiple OSA ports into a single logical pipe. Link aggregation enables greater throughput and network load-balancing at the Layer 2 level when communicating with a physical switch. This technology extends to multiple systems in a z/VM SSI through Inter-VSWITCH link aggregation groups.

For more information about virtual networking options in z/VM, see z/VM: Connectivity.

2.5.4 HyperSwap

HyperSwap is a storage HA solution for IBM Storage subsystems (such as the DS8000) that enables switching logical units between two storage subsystems to help ensure continuous operation. HyperSwap is a HA feature that provides single-site or dual-site, active-active access to a volume. This function helps ensure continuous data availability if there is a hardware failure, power failure, connectivity issues, or other unplanned outages. It offers a robust DR solution with a minimal recovery point objective (RPO) and recovery time objective (RTO).

For more information about HyperSwap, see 3.2.5, “HyperSwap function” on page 74.

2.5.5 Multipathing

Multipath I/O provides failover and might improve performance. You can configure multiple physical I/O paths between server nodes and storage arrays as a single multipath device. Multipathing aggregates the physical I/O paths, which create a new device that consists of the aggregated paths.

Multipathing provides I/O failover and path load sharing for multipathed block devices. In Linux, multipathing is implemented with multi-path tools that provide a user-space daemon for monitoring and an interface to the device mapper. The device-mapper provides a container for configurations and maps block devices to each other.

A single SCSI device (or a single zFCP[5] unit) constitutes one physical path to storage. The multipath user-space configuration tool scans `sysfs` for SCSI devices and then groups the paths into multipath devices. This mechanism that automatically puts each detected SCSI device underneath the correct multipath device is called *coalescing*.

Use a multipath setup to access SCSI storage in an Fibre Channel Storage Area Network (FC SAN). The multipath device automatically switches to an alternative path if there is an interruption on the storage system controllers or due to maintenance on one path.

The multipath daemon has default configuration entries for most Storage Systems, so you need to do only basic configuration for these systems.

Figure 2-11 shows an example of a multipath implementation with Linux.

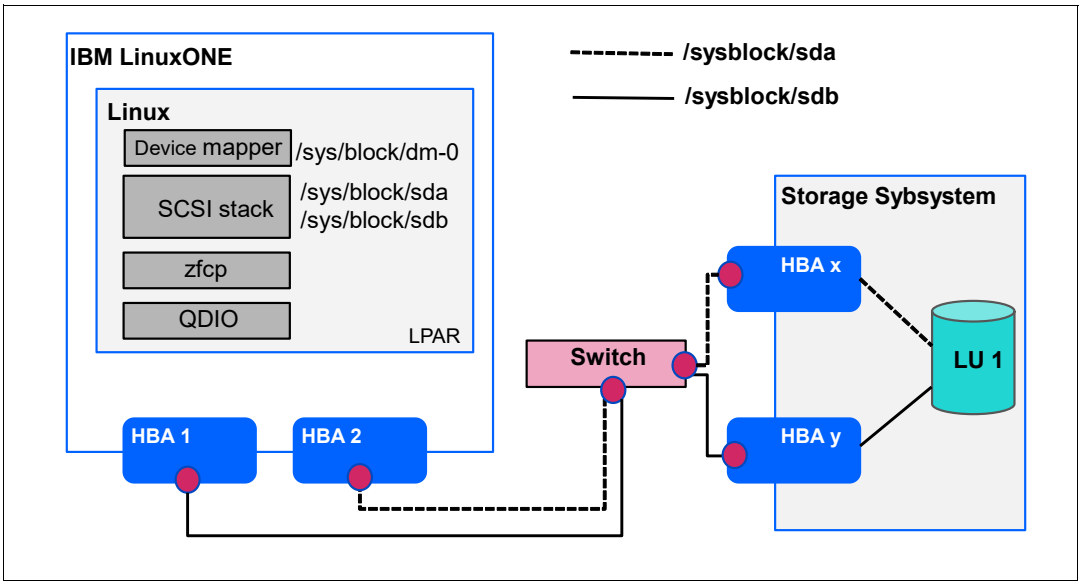


Figure 2-11 Example of a multipath implementation with Linux

For more information about how to access, configure, and use FCP Multipathing with the Linux kernel, see IBM Documentation.

[5] The zFCP device driver is a low-level driver or host bus adapter driver that supplements the Linux SCSI stack. For more information, see
https://www.ibm.com/docs/en/linux-on-z?topic=channel-what-you-should-knowok.

2.6 Applications

This section covers applications in the IBM LinuxONE platform and introduces container technology, containerization, and containerized workloads. Also covered are middleware, middleware types, independent software vendors (ISVs) applications, and supported databases.

2.6.1 Containers

Container images become containers at run time. Containerized software is available for Linux or other platforms such as Windows or cloud, and it always runs the same regardless of the infrastructure. Containers isolate software from its environment and help ensure that the software works uniformly despite differences, for example, between development and staging.

Containers are standard units of software that package code and all its dependencies so that the application runs quickly and reliably from one computing environment to another one. Container images are a lightweight, stand-alone, and executable package of software that includes everything that is needed to run an application:

- Code
- Run time
- System tools
- System libraries
- Settings

Containers are a way to bundle and run applications. In a production environment, you must manage the containers that run the applications and help ensure that there is no downtime. For example, if a container goes down, another container must start. Kubernetes provides a framework to run distributed systems resiliently. It manages application scaling and failover, and deployment patterns.

Figure 2-12 shows a traditional virtualization implementation compared to a containerized implementation that is based on a container engine.

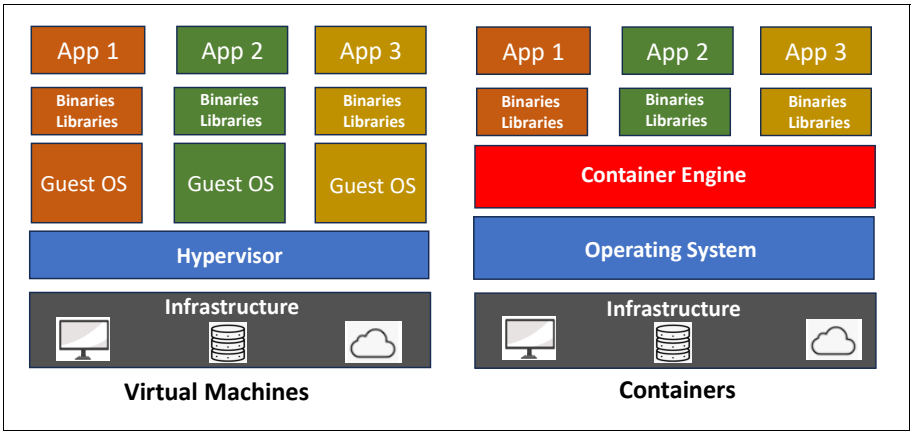


Figure 2-12 Virtualized versus containerized environments

2.6.2 Containerization

Containerization is the packaging of software code with only the OS libraries and the dependencies that are required to run the code to create a single lightweight executable file (container) that runs consistently on any infrastructure. Containers are more portable and resource-efficient than a VM, and they have become the *de facto* compute units of modern, cloud-native applications.

With containerization, developers can create and deploy applications faster and more securely. With traditional methods, code is developed in a specific computing environment that, when transferred to a new location, often results in bugs and errors. For example, when a developer transfers code from a desktop computer to a VM. Containerization eliminates this problem by bundling the application code with the related configuration files, libraries, and dependencies that are required for it to run. This single package of software or “container” is abstracted away from the host OS, so it stands alone and becomes portable, and able to run across any platform or cloud free of issues.

Containers are often referred to as “lightweight”, which means that they share the machine’s OS kernel and do not require the overhead of associating an OS within each application. Containers are inherently smaller in capacity than a VM and require less start-up time, which enables more containers to run with the same compute capacity as a single VM. This drives higher server efficiencies and reduces server and licensing costs.

For more information, see Containers in the enterprise.

2.6.3 Containerized workloads

Containerized workloads on IBM LinuxONE benefit from the platform's reliability, security, and scalability. With advanced virtualization capabilities and integration with container orchestration platforms, LinuxONE provides an excellent solution for running diverse containerized applications, which include Linux-based workloads, microservices, cloud-native applications, and hybrid cloud deployments. For more information, see 2.3.7, “Container platforms” on page 34.

2.6.4 Middleware

Middleware is software that enables one or more kinds of communication or connectivity between applications or application components in a distributed network. By making it simpler to connect applications that are not designed to connect with one another and providing functions to connect them in intelligent ways, middleware streamlines application development and speeds time to market.

Middleware enables developers to build applications without having to create a custom integration every time that they need to connect to application components (services or microservices), data sources, computing resources, or devices. It does this by providing services that enable different applications and services to communicate by using common messaging frameworks, such as JSON, REST, XML, SOAP, or web services. Typically, middleware also provides services that enable components that are written in multiple languages, such as Java, C++, PHP, and Python to talk with each other.

Middleware services

In addition to providing work-saving interoperability, middleware also includes services that help developers complete the following tasks:

- Configure and control connections and integrations.

  Based on information in a client or front-end application request, middleware can customize the response from the back-end application or service. In a retailer's e-commerce application, middleware application logic can sort product search results from a back-end inventory database by nearest store location, which is based on the IP address or location information in the HTTP request header.

- Secure connections and data transfer.

  Middleware typically establishes a secure connection from the front-end application to back-end data sources by using Transport Layer Security (TSL) or another network security protocol. Middleware can provide authentication capabilities, which challenge front-end application requests for credentials (username and password) or digital certificates.

- Manage traffic dynamically across distributed systems.

  When application traffic spikes, enterprise middleware can scale to distribute client requests across multiple servers, whether on-premises or in the cloud. Concurrent processing capabilities can help prevent problems when multiple clients try to access the same back-end data source simultaneously.

Types of middleware

Here are the types of middleware:

- Message-oriented middleware (MOM)

  Enables application components by using different messaging protocols to communicate and exchange messages. In addition to converting or transforming messages between applications, MOM manages the routing of the messages so that they always get to the proper components in the proper order. Examples of MOM include message queues and message brokers.

- Remote procedure call (RPC) middleware

  Enables one application to trigger a procedure in another application that runs on the same computer or on a different computer or network as thought both applications are part of the same application on the same computer.

- Data or database middleware

  Simplifies access to and interaction with back-end databases. Typically, database middleware is some form of SQL database server.

- API middleware

  Provides tools that developers can use to create, expose, and manage APIs for their applications so that other developers can connect to them. Some API middleware includes tools for monetizing APIs, which enables other organizations to use them at cost. Examples of API middleware include API management platforms, API gateways, and API developer portals.

- Object request broker (ORB) middleware

  Acts as a broker between a request from one application object or component and the fulfillment of that request by another object or component on the distributed network. ORBs operate within the Common Object Request Broker Architecture (CORBA), which enables one software component to make a request of another one without knowing where the component is hosted or what its UI looks like. “Brokering” handles this information during the exchange.

- Transactional middleware

  Provides services to support the execution of data transactions across a distributed network. The best-known transactional middleware is transaction processing monitors (TPMs), which help ensure that transactions proceed from one step to the next, such as running the data exchange, or adding, changing, or deleting data where needed, through to completion.

- Asynchronous data streaming middleware

  Replicates a data stream in an intermediate store, which enables data sharing between multiple applications. Apache Kafka is one of the best-known examples of middleware for real-time data streaming.

- Device middleware

  Provides a focused set of integration and connectivity capabilities for developing apps for a specific mobile OS.

- Portal middleware

  Provides tools and resources for integrating content and capabilities from various related applications on a single screen to create a single, composite application.

- Robotics middleware

  Simplifies the process of integrating robotic hardware, firmware, and software from multiple manufacturers and locations.

2.6.5 Independent software vendor applications

ISVs develop applications that run and expand on an existing solution. IBM works closely with ISVs to help ensure applications that are developed for LinuxONE platforms provide more value to clients.

To explore the list of ISVs for IBM LinuxONE, see the IBM LinuxONE Partner Network (LPN) program. This program helps ISVs easily port, certify, and deploy applications on IBM LinuxONE. ISVs can also gain access to go-to-market resources and a rich set of learning tools for skill development 1.

2.6.6 Databases

IBM LinuxONE supports various databases. IBM LinuxONE servers are designed for secure data serving and can run multiple Linux-based workloads that include Oracle Database 19c, Oracle WebLogic Server, open source, blockchain, and other Linux-based commercial software.

This article provides an overview of IBM LinuxONE support for open source databases like MongoDB PostgreeSQL and MariaDB.

2.7 Management

Managing an IBM LinuxONE environment can be accomplished by using several tools. Which tools you use depend on the virtualization environment. This section covers IBM Operations Manager, z/VM Centralized Manager, and IBM Infrastructure Suite for z/VM and Linux. The available KVM and Container Management tools are also described.

2.7.1 z/VM Hypervisor Management

This section describes the following topics:

- IBM Operations Manager for z/VM
- z/VM Centralized Service Management
- IBM Infrastructure Suite for z/VM and Linux

IBM Operations Manager for z/VM

IBM Operations Manager for z/VM supports automated operational monitoring and management of z/VM VMs and Linux guests. It can help you address issues before they impact your service-level agreements (SLAs). Systems programmers and administrators can automate routine maintenance tasks in response to system alerts. Users can easily debug problems by viewing and interacting with consoles for service machines and Linux guests. Operators can better interpret messages and determine corrective actions.

Operations Manager for z/VM has the following benefits:

- Automated response to issues: Automated actions can be created to address console messages on z/VM service machines and Linux guests.
- Advanced monitoring: Monitoring for z/VM page space and spool files.
- Automated tasks: Tasks can be performed at specific times by using scheduled events without manual intervention.
- Rule-specific issues checking: z/VM and Linux guests are monitored based on predefined rules that you create.

z/VM Centralized Service Management

Managing multiple remote and local systems while coordinating levels of service among these systems can be difficult. New with z/VM 7.2 is a service management tool that is called z/VM Centralized Service Management (z/VM CSM).

When using CSM, one system is designated as the principal system. This system uses the Shared File System (SFS) to manage service levels for a set of defined managed systems regardless of their geographic location. The new **SERVMGR** command uses Virtual Machine Serviceability Enhancements Staged/Extended (VMSES/E) commands to apply service and local modifications, build serviced content, and drive the transport of the packaged service to the managed systems.

IBM Infrastructure Suite for z/VM and Linux

IBM Infrastructure Suite for z/VM and Linux is a single solution that provides multiple tools to monitor and manage z/VM and Linux virtual servers on IBM Z systems as part of both traditional and cloud infrastructure. It supports backup and recovery of the entire system at a reduced price.

The capabilities of IBM Infrastructure Suite for z/VM and Linux provide comprehensive insight to efficiently control and support your IBM z/VM and Linux on IBM Z systems environments.

Here are the products that are included with IBM Infrastructure Suite for z/VM and Linux:

- IBM Tivoli® OMEGAMON® XE for Messaging on z/VM and Linux
- IBM Spectrum® Protect Extended Edition
- IBM Operations Manager for z/VM
- IBM Backup and Restore Manager for z/VM
- IBM Tape Manager for z/VM (optional)
- IBM Cloud Infrastructure Center (optional)

2.7.2 KVM Hypervisor Management

Various tools exist for managing KVMs efficiently:

- Libvirt

  Libvirt is a toolkit that provides a consistent and stable API for managing various virtualization solutions, including KVM. It offers command-line utilities (virsh) and APIs for management. For more information, see KVM Management Tools.

- Virt-manager

  Virt-manager is a desktop application for managing KVMs. It provides a GUI to create, view, modify, and manage VMs. For more information, see Good GUI for KVM.

- Cockpit Web Console

  Cockpit is a web-based server management tool that includes support for managing KVMs. It offers a web interface for VM management. For more information, see Tecmint.

- WebVirtMgr:

  WebVirtMgr is a web-based interface for managing KVMs. It provides a browser-accessible platform to create and control VMs. For more information, see Reddit.

2.7.3 Container management

Container management automates the creation, deployment, and scaling of containers. It also facilitates the organization of containers. Container management enables systems to work more efficiently, especially when the number of containers becomes too large for the IT team to handle. With Container management, IT teams can keep their environment more secure, and developers can explore its flexibility to create and deploy new apps and services.

Docker container management software and Kubernetes container management facilitate the orchestration, security, and networking tasks. Container management software systems are available as both open-source and proprietary commercial products.

2.7.4 Kubernetes

Kubernetes is an open-source platform for managing containerized workloads and services that facilitates both declarative configuration and automation. Kubernetes is portable, extensible, and has a large and rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

Here are some of the Kubernetes features:

- Auto scaling
- Storage orchestration
- Self-healing
- Multi-cloud and hybrid cloud support

- Load-balancing
- Applications rollout (up-to-date)
- Community support

For more information about Kubernetes, see Kubernetes Documentation.

2.7.5 Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform is an example of container technology. Red Hat OpenShift Container Platform is a trusted Kubernetes enterprise platform that supports modern, hybrid-cloud application development and provides a consistent foundation for applications anywhere across physical, virtual, private, and public clouds.

Red Hat OpenShift Container Platform provides an abstraction layer with the same function regardless of the cloud deployment model (on-premises, public cloud, private cloud, or hybrid cloud) and hardware architectures (x86, s390x, ppc64le, or ARM), and it is used as the foundation for how IBM distributes software in a containerized format.

Figure 2-13 shows the deployment options for Red Hat OpenShift Container Platform on IBM LinuxONE.

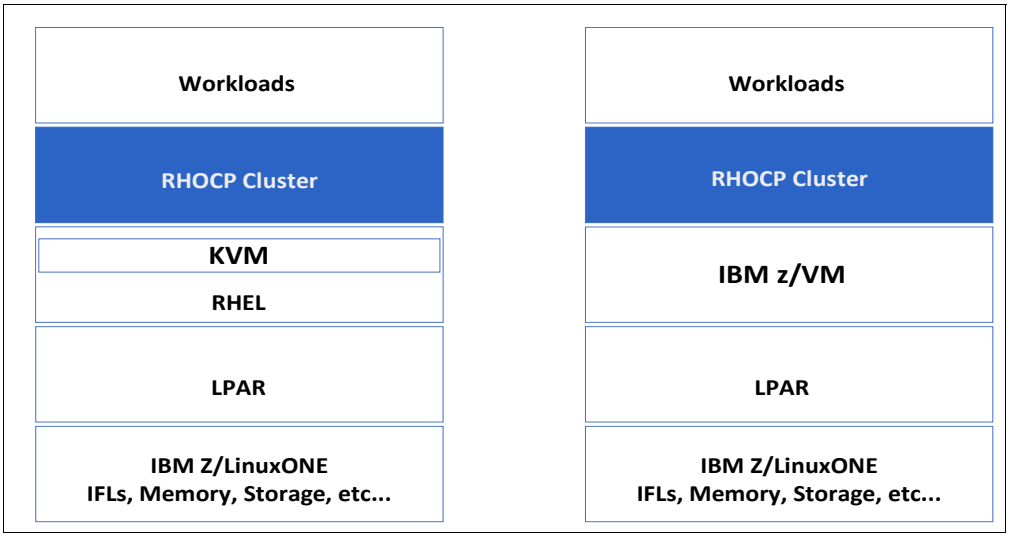


Figure 2-13 Deployment options for Red Hat OpenShift Container Platform on IBM LinuxONE

2.8 Automation

The IBM LinuxONE platform can leverage several automation tools and processes, depending on the chosen virtualization method. This section describes data replication, Copy Services Manager (CSM), automation with Linux HA, Ansible, IBM Cloud Infrastructure Center, and GDPS.

2.8.1 Synchronous data replication

Peer-to-Peer Remote Copy (PPRC) is a hardware solution that provides rapid and accurate DR and a solution to workload movement and device migration. Updates that are made on the primary DASD volumes are synchronously shadowed to the secondary DASD volumes. The local storage subsystem and the remote storage subsystem are connected through a communications link that is called a PPRC path. FCP is the protocol that is used to copy data by using PPRC.

PPRC is configured in the storage subsystems by using local administration capabilities. For an IBM DS8000, you can use the following tools to configure PPRC:

- Hardware Management Console (HMC)
- HMC Command-Line Interface (CLI)
- CSM (For more information, see 2.8.2, “Copy Services Manager” on page 48.)

2.8.2 Copy Services Manager

CSM controls copy services in storage environments. Copy services are features that are used by Storage Systems such as IBM DS8000 to configure, manage, and monitor data-copy functions. Copy services include the point-in-time function, IBM FlashCopy®, MM, GM, and Metro Global Mirror (MGM). CSM can automate the administration and configuration of these services, and monitor and manage copy sessions.

You can use CSM to complete the following data replication tasks and help reduce the downtime of critical applications:

- Plan for replication when you are provisioning storage.
- Keep data on multiple related volumes consistent across Storage Systems for a planned or unplanned outage.
- Monitor and track replication operations.
- Automate the mapping of source volumes to target volumes.

Starting with DS8000 8.1, CSM also comes preinstalled on the HMC. Therefore, you can enable the CSM software that is already on the hardware system, which results in less setup time and eliminates the need to maintain a separate server for Copy Services functions.

CSM can also run on Linux on IBM Z, and it uses FICON to connect and manage IBM Storage Systems count-key data (CKD) volumes.

2.8.3 Automation with Linux HA components

Using open-source components, you can automate the resiliency of an environment to recover or switch workloads and help ensure continuous operation.

The major open-source components are Pacemaker and Corosync, which are part of Red Hat Enterprise Linux (RHEL) HA, SUSE HA, and Ubuntu HA.

Pacemaker

Pacemaker is an open-source HA cluster resource manager software that runs on a set of nodes. Together with Corosync, which is an open-source group communication system that provides ordered communication delivery, cluster membership, quorum enforcement, and other features among the nodes, it helps detect component failures and orchestrates necessary failover procedures to minimize interruptions to applications.

Pacemaker can supervise and recover from failures within a cluster. Here are the components of Pacemaker architecture:

- Cluster Information Base (CIB):
  - CIB is the Pacemaker information daemon. It uses XML to distribute and synchronize the current configuration and status from the Designated Coordinator (DC). The DC node is assigned by Pacemaker to store and report cluster state and actions to the other nodes by using the CIB. There is a CIB in each host instance.
  - The XML list of behaviors, which are directed by the resource manager, informs the policy engine.
- Cluster Resource Management daemon (CRMd):
  - Pacemaker uses this daemon to route cluster resource actions. Resources that are managed by this daemon can be queried, moved, instantiated, and changed when needed.
  - Communicates with the Local Resource Manager daemon on each cluster.
  - The Local Resource Manager receives instructions from the CRMd and passes requests along to local resource agents (VirtualDomain, FileSystem, and MailTo) for general operations.
  - Defines one Designated Focal Point for the cluster.
  - The other hosts receive data from CRMd through Corosync.
- Shoot the Other Node in the Head (STONITH):
  - A Pacemaker fencing agent that detects whether it loses contact with one of the nodes in the cluster.
  - If Pacemaker thinks that a node is down, STONITH forces it offline.
  - The Pacemaker fencing is implemented by STONITH.
  - STONITH shuts down and fences nodes, which removes them from the cluster to maintain data integrity.
- Corosync:
  - Corosync is a component/daemon that handles core membership enrollment and the required communication with the members for cluster HA for any Linux instance that is enrolled in an HA cluster.
  - Corosync is a required component and a daemon in a Linux HA cluster.
  - Manages quorum rules and determination.
  - Starts and stops the VMs.

Policy Engine

Policy Engine is a software component that helps manage policies for clusters. Policy Engine defines what users can do on a cluster and helps ensure that clusters can communicate. Anytime that a Kubernetes object is created, a policy evaluates and validates or mutates the request. Policies can apply across a namespace or different pods with a specific label in the cluster. Kubernetes policy engines block objects that might harm or affect the cluster if they do not meet the policy's requirements, and it takes that list of behaviors and maps them to cluster's current state.

OpenSAF checkpoint APIs

OpenSAF is a checkpoint service that provides mechanisms for checkpointing and restoring an application state. It provides a series of APIs to help applications register for checkpoints and to perform the restoration of applications.

- Service Availability Forum (SAF): Creates the Service Availability Specifications.
- OpenHPI: The Hardware Platform Interface (HPI) abstracts the differences between hardware implementations, which provide a uniform interface to the hardware features.
- OpenAIS: The Application Interface Specification (AIS) specifies an interface that applications use to exchange information with the service availability middleware (Cluster Resource Management).

Heartbeat

The heartbeat of a node in a cluster is a signal that is sent between nodes to indicate that they are still alive and functioning properly. It is used to detect when a node fails and initiate failover procedures. Sometimes, the heartbeat is used to monitor the health of a node and initiate maintenance actions.

- Uses messaging between nodes to make sure that they are alive and available.
- Determines whether an action is required when the heartbeat stops after a certain number of tries.

Cluster Glue

Cluster Glue is a set of libraries, tools, and utilities that are used in the heartbeat and Pacemaker cluster stack. In essence, Cluster Glue is the parts of the cluster stack that do not fit in anywhere else. Basically, it is everything that is not messaging layer and not a resource manager.

Resource agents

Resource agents are scripts that enable Pacemaker to manage any service that it is unknown to Pacemaker. The agents contain the logic for what to do when the cluster wants to start, stop, or check the health of a service.

- A resource agent is an executable file that manages a cluster resource. No formal definition of a cluster resource exists, other than “anything that a cluster manages is a resource”. Cluster resources can be as diverse as IP addresses, file systems, database services, and entire VMs.
- Resource agents run in clustered systems or remotely.
- Resource agents can start, stop, or restart services.
- Quorum helps ensure that the majority of a cluster agrees on what the resource is. Votequorum is the interface that members use to agree.
- Resource agents also provide messaging for applications coordinating or operating across multiple members of a cluster.

2.8.4 Automation with Ansible: Overview

Ansible is an open-source automation tool that simplifies IT tasks, which include configuration management, application deployment, and orchestration. You can use it to automate routine tasks, which reduce the risk of human error and improves system resiliency. LinuxONE is designed for HA and security. Ansible complements these features by automating repetitive tasks, helping ensure consistent configurations, and enabling rapid responses to system events.

With Red Hat Ansible Automation Platform running on IBM LinuxONE, you have an enterprise-supported solution for automation on various levels that can span all platforms in the enterprise.

2.8.5 IBM Cloud Infrastructure Center automation

IBM Cloud Infrastructure Center is a software platform for managing the infrastructure of private clouds on IBM LinuxONE. The IBM Cloud Infrastructure Center is an IaaS offering that delivers an industry-standard user experience for the IaaS management of non-containerized and containerized workloads IBM LinuxONE.

2.8.6 Resiliency automation for the infrastructure

With the requirements for resiliency, different businesses have different requirements for their operational model, such as business continuity (BC) or continuous availability (CA).

- Business continuity: Continuously operates and masks planned outages from users. It employs nondisruptive hardware and software changes, nondisruptive configuration, and software coexistence.
- Continuous availability: Delivers nondisruptive service to the user 24/7 (no planned or unplanned outages).

The final goal is to provide CA, which means that it is important to have the capabilities of problem and failure detection and recovery automation processes if one of the layers or components are malfunctioning and need recovery actions.

2.8.7 Automation with GDPS

GDPS solutions are designed to accomplish the following tasks:

- Deliver end-to-end application and data availability within a site or across multiple sites.
- Automate recovery procedures for near-CA and DR.
- Provide rapid RPO and RTO.
- Provide multiple, secure point-in-time copies of production data to help protect them from logical corruption threats.
- Monitor systems and storage that support open and IBM copy technology architectures, which include IBM, Hitachi Vantara, and MM (only for non-IBM storage).
- Simplify management tasks with a central point of control.

Multi-site environment

IBM GDPS technology provides a total BC solution for multi-site environments. The technologies also provide a collection of end-to-end automated DR solutions on the IBM LinuxONE platform, with each one addressing a different set of IT resiliency goals that can be tailored to meet the recovery objectives for your business.

Depending on the solution, GDPS supports the following tasks:

- Managing and monitoring the remote copy environment
- Managing and monitoring the production environment
- Data that spans more than one platform, including z/OS, z/VM, KVM, and Linux on IBM Z
- Multiple disk subsystem vendors

- Transparent and near-continuous application availability from disk failures
- Automated and transparent failover and failback

GDPS extends the resource sharing, workload balancing, and CA benefits of a parallel environment. It also enhances the capability of an enterprise to recover from disasters and other failures, and to manage planned exception conditions. As a result, you can achieve your own CADR goals.

GDPS is a collection of several offerings, each addressing a different set of IT resiliency goals that can be tailored to meet the RPO and RTO for your business.

Each GDPS offering uses a combination of system and storage hardware-based or software-based replication and automation, and clustering software technologies:

- Metro
- Metro HyperSwap Manager
- IBM GDPS Virtual Appliance
- Extended Disaster Recovery (xDR)
- GM
- MGM
- CA

The following sections describe the applicable LinuxONE applicable technologies: Metro, GDPS Virtual Appliance, xDR, GM, and MGM.

GDPS Metro

A near-CADR solution across two sites that are separated by metropolitan distances. The solution is based on the MM synchronous disk mirroring technology.

Provides CADR and production system/sysplex resource management capabilities. Based on the MM synchronous disk mirroring technology, it can achieve an RPO of zero. Typically, the recovery time is less than one hour (RTO < 1 hour) after a complete site failure. Metro also supports HyperSwap to provide near-continuous disk availability after a disk failure.

GDPS Global Mirror

A DR solution across two regions that are separated by virtually unlimited distance. The solution is based on the IBM System Storage GM technology, which is a disk subsystem-based asynchronous form of remote copy.

GDPS Metro Global Mirror

A 3-site or a symmetrical 4-site configuration is supported:

- MGM 3-site

  A 3-site solution that provides CA across two sites within metropolitan distances in one region and DR to a third site in a second region at virtually unlimited distances. It is based on a combination of the MM and GM technologies.

- MGM 4-site

  A symmetrical 4-site solution that is similar to the 3-site solution in that it provides CA within a region and DR across regions. Also, in the 4-site solution, the two regions are configured symmetrically so that as much CADR protection is provided, no matter which region production runs in.

GDPS Virtual Appliance

The GDPS Virtual Appliance is a software appliance that is a self-contained OS, application, middleware, APIs, and GUI in one IBM LinuxONE LPAR software image.

GDPS Virtual Appliance requires that the LinuxONE configuration has a PU that is characterized as a CP. This required CP is ordered as a Feature Code and does not incur any utilization charges.

GDPS Virtual Appliance supports both planned and unplanned situations, which help to maximize application availability and provide BC. In particular, a GDPS Virtual Appliance solution can deliver the following capabilities:

- Near-CA solution
- DR solution across metropolitan distances
- RTO of less than an hour
- RPO of zero

The GDPS Virtual Appliance can start and stop z/VM or KVM hypervisors, and thus all Linux guests that are running on these hypervisors. It is a fully integrated software solution for providing CADR protection to Linux on IBM Z, zVM, and KVM. If your Linux on IBM Z production workloads have CADR requirements, then the GDPS Virtual Appliance can help you meet those requirements.

A GDPS Virtual Appliance environment is typically spread across two data centers (Site1 and Site2)[6] where the primary copy of the production disk is normally at Site1. The GDPS Virtual Appliance must have connectivity to all the Site1 and Site2 primary and secondary devices that it manages. For availability reasons, the GDPS Virtual Appliance runs at Site2 on a local disk that is not mirrored with MM. This approach provides failure isolation for the appliance system to help ensure that it is not impacted by failures that affect the production systems and remains available to automate any recovery action.

[6] GDPS Virtual Appliance may also be implemented in a 2-system configuration that is installed at a single site.

Figure 2-14 shows a typical GDPS Virtual Appliance implementation.

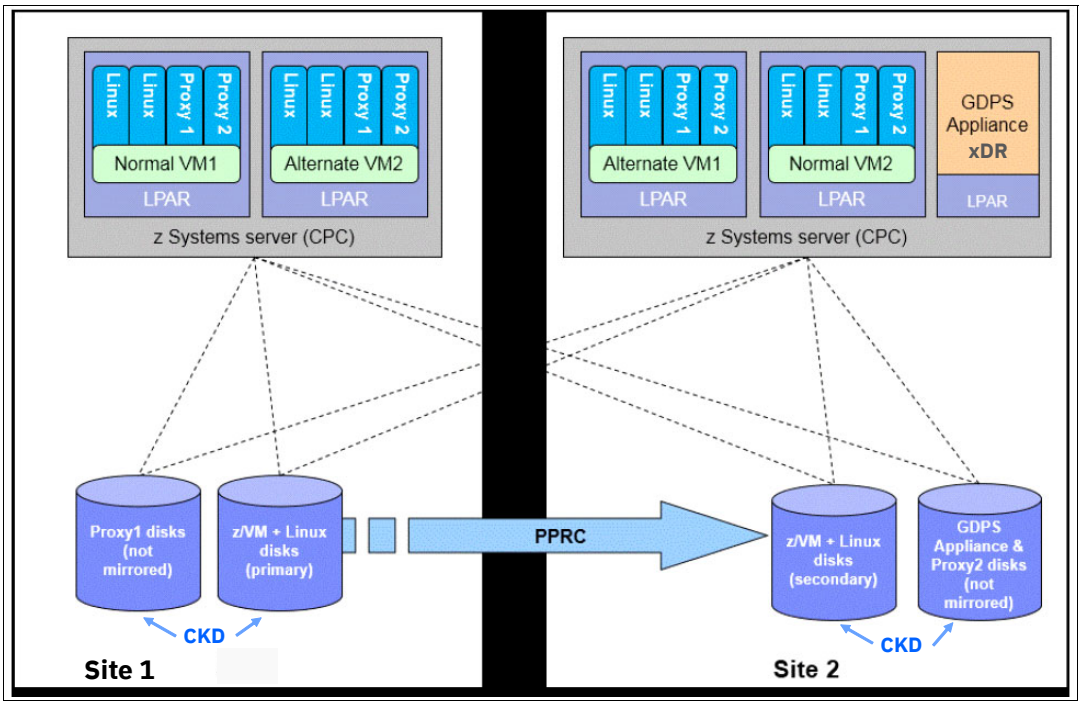


Figure 2-14 Typical GDPS Virtual Appliance implementation

GDPS Virtual Appliance for LinuxONE

GDPS Virtual Appliance provides the following functions for LinuxONE:

- Coordinated takeover in unplanned cases, for example, recovery from a node failure
- Coordinated takeover in planned cases, for example, maintenance
- Coordinated Planned HyperSwap (PHS) and Unplanned HyperSwap (UHS) (with z/VM)
- Freeze support
- LAN with z/VM SSI
- Management of all disks to control mirroring
- Management of all LPARs to start and stop them.
- Graceful shutdown and startup of z/VM and Linux guests

GDPS Virtual Appliance HyperSwap function

The GDPS Virtual Appliance delivers a powerful function that is known as HyperSwap. HyperSwap can swap from using the primary devices in a mirrored configuration to using what were the secondary devices. This process is transparent to the production systems and applications that use these devices.

Without HyperSwap, a transparent disk swap is not possible. In this case, all systems that use the primary disk must shut down (or might fail, depending on the nature and scope of the failure) and must restart by using the secondary disks. Disk failures are often a single point of failure (SPOF) for the entire production environment.

Extended Disaster Recovery: Multiplatform Resiliency with GDPS Virtual Appliance

GDPS Virtual Appliance includes the Metro and the xDR capabilities. The xDR function extends the near-CADR capabilities that are provided by Metro to other platforms or OSs that are running on LinuxONE servers.

With xDR, you can communicate with and manage z/OS Proxy (z/OS systems monoplexes outside of the sysplex), z/VM, KVM, Secure Service Container (SSC) (IBM Db2® Analytics Accelerator (IDAA)), and Linux in an LPAR. To access these capabilities, run System Automation for Multiplatforms (SA MP) by using a separately licensed xDR feature.

The proxy nodes communicate commands from and to z/VM, monitor the z/VM environment, and communicate status and failure information back to the GDPS Virtual Appliance.

Extended Disaster Recovery for z/VM with GDPS Virtual Appliance

In each xDR managed z/VM system, you must configure two special Linux guests, which are known as the *proxy* guests. One proxy node is configured on the Site1 disk and the other is configured on the Site2 disk. The proxies are guests that are dedicated to providing communication and coordination with the GDPS Virtual Appliance. They must run SA MP by using a separately licensed xDR feature.

Extended Disaster Recovery for KVM with GDPS Virtual Appliance

Multiplatform Resiliency for KVM uses the xDR protocol to communicate with an xDR KVM proxy to send **`libvirt`** commands, so xDR must be enabled to support KVM.

The xDR KVM Proxy is delivered as a Linux RPM for SUSE Linux Enterprise Server or RHEL, or as a DEB package for Ubuntu. The proxy guest serves as the middleware that communicates commands from and to KVM, monitors the KVM environment, and communicates status information back to the Metro controlling system.

KVM does not provide a HyperSwap function. However, Metro coordinates PHS and UHS for Linux under z/VM and Linux under KVM CKD disks to maintain data integrity and control the shutdown and “restart in place” of the KVM LPARs. For disk or site failures, Metro provides a coordinated freeze for data consistency on CKD disks across KVM, z/VM, and Linux LPARs.

Extended Disaster Recovery for z/OS with IBM Z

In a scenario where a client already has a z/OS Parallel Sysplex environment running, Metro provides a function that called *Metro Multiplatform Resiliency for IBM Z*, also known as xDR.

This function is especially valuable for customers who share data and storage subsystems between z/OS and Linux z/VM guests on IBM Z or SUSE Linux running natively on IBM Z LPARs, for example, an application server running on Linux on IBM Z and a database server running on z/OS.

With a multi-tiered architecture, you must provide a coordinated, near-CADR solution for both z/OS and Linux.

GDPS Metro can provide this capability when Linux is running as a z/VM guest or natively. Using the HyperSwap function so that the virtual device that is associated with one real disk can be swapped transparently to another disk, HyperSwap can be used to switch to secondary disk storage subsystems that are mirrored (MM). If there is a hard failure of a storage device, GDPS Metro coordinates the HyperSwap with z/OS for CA spanning the multi-tiered application. HyperSwap is supported for ECKD and xDR managed FB disks.

For site failures, GDPS starts the freeze function for data consistency and rapid application restart without needing data recovery. HyperSwap can also be helpful in data migration scenarios to enable applications to migrate to new disk volumes without requiring them to be quiesced.

When using an ECKD formatted disk, GDPS Metro can provide the reconfiguration capabilities for the Linux on IBM Z servers and data the same as for z/OS systems and data.

To support planned and unplanned outages, these functions are extended to KVM with GDPS 4.5 and later, which provides recovery actions, such as the following examples:

- “Restart in place” of the failing OS images.
- Heartbeat checking of Linux guests.
- Disk error detection.
- Data consistency with freeze functions across z/OS and Linux.
- Site takeover/failover of a complete production site.
- Single point of control to manage disk mirroring configurations.
- Coordinated recovery for planned and unplanned events.

More support is available for Linux running as a guest under z/VM:

- “Restart in place” of failing OS images.
- Order Linux node or cluster startup and shutdown.
- Coordinate PHS and UHS of disk subsystems, which are transparent to the OS images and applications that use the disks.
- Transparent disk maintenance and failure recovery with HyperSwap across z/OS and Linux applications.

GDPS MGM

GDPS provides DR and production system and sysplex resource management capabilities. Based on GM synchronous disk mirroring technology, it can achieve data loss down to 3 seconds (RPO = 3 seconds) and RTO < 1 hour. GDPS Global supports two sites that are separated by virtually unlimited distances with minimal impact to user response times. Three-site and four-site configurations are available, such as combining GDPS MM with GM to form GDPS MGM.

MGM has the following features:

- Down to zero data loss
- HyperSwap for disk availability
- Protection from regional events
- Minimal impact to user response time

For more information about GDPS offerings, see *IBM GDPS: An Introduction to Concepts and Capabilities*, SG24-6374.

Table 2-1 shows the capabilities of various GDPS components that are applicable to LinuxONE resiliency scenarios.

Table 2-1 GDPS components that are applicable to LinuxONE

Feature	GDPS Metro[a]	GDPS Virtual Appliance[b]	GDPS GM
CA	Yes	Yes	No
DR	Yes	Yes	Yes
CADR protection against multiple failures	Yes	No	No
Max Supported distance	200 KM (fiber)	200 KM (fiber)	Virtually unlimited

Feature	GDPS Metro[a]	GDPS Virtual Appliance[b]	GDPS GM
Span of control	Both sites	Both sites	Disk: both sites Recovery site: CBU or LPARs
GDPS scripting	Yes	Yes	Yes
Monitoring, alerting, and health checks	Yes	Yes No health checks	Yes
Query services	Yes	No	Yes
Fixed-block disk	Yes	No	Yes
z/OS equivalent function for Linux on IBM Z	Yes (Linux as z/VM Guest)	Yes (Linux as z/VM Guest)	Yes
GDPS GUI	Yes	Yes	Yes

a. GDPS xDR is part of GDPS Metro.
b. GDPS Virtual Appliance includes GDPS Metro capabilities.

2.8.8 Infrastructure placement

By design, the IBM LinuxONE platform is a highly resilient system. Through RAS design principles, the LinuxONE architecture has built-in self-detection, error correction, and redundancy. The LinuxONE platform reduces SPOFs to deliver the best reliability of any enterprise system in the industry. Transparent processor sparing and dynamic memory sparing enable concurrent maintenance and seamless scaling without downtime.

A single-system environment has redundancy in the hardware and defines resiliency based on the application and service availability.

For resiliency reasons, sensitive enterprises are using IBM LinuxONE systems in a redundant deployment in different topologies.

There are three deployments for resilience:

- Hot-Cold (Active-Passive in a distributed environment)
- Hot-Warm (Active-Idle in a distributed environment)
- Hot-Hot (Active-Active in a distributed environment)

Characteristics of a Hot-Cold (Active-Passive) deployment

This deployment has two data centers with at least one LinuxONE server per data center and its associated storage servers, network infrastructure, and services.

The main data center, the PROD data center, is the one where production runs. The DR data center is the second data center.

In an active-passive environment, clients want cost-efficiency, which means to keep the system in the DR data center shut down, and with active storage servers in PROD and DR for permanent data replication. In a disaster, the system in the DR data center activates and PROD runs in the DR data center.

Storage mirroring can be implemented at the storage server level with MM (synchronous replication) or with GM (asynchronous replication).

Characteristics of a Hot-Warm (Active-Idle) deployment

In this deployment, the PROD environment runs the entire production workload, but the DR data center has a started system with some if its basic setup active, such as hypervisors, replication software for data, or special software for automatic failover. The storage servers in PROD and DR are set up for replication and mirroring of all data from PROD.

Similar to the first case, storage mirroring can be implemented on the storage server level and can be MM or GM.

For container workloads, the mirroring can also be implemented on a logical level if, on the DR site, the storage replication software is actively running, such as IBM Storage Fusion Data Foundation or IBM Storage Scale servers.

Characteristics of a Hot-Hot (Active-Active) deployment

The most effective resiliency can be reached in an environment where two sites are active concurrently and are set up to take over the entire PROD workload whenever one site is failing or suffering a disaster.

For this environment, you must have a storage infrastructure that can share the storage between the two sites if the same applications are running concurrently at both sites.

This scenario depends on the type of workload that is used, whether it is a traditional VM-based workload, container-based workloads, or a combination of both.

2.9 Building the related deployment model

There are different approaches for creating a deployment model with IBM LinuxONE. The following section presents best practices for leveraging hardware and virtualization capabilities. For more information, see Chapter 3, “Deployment scenarios for resiliency” on page 61.

2.9.1 Best practices for hardware and virtualization

Here are some best practices for hardware and virtualization:

- For a resilient environment, consider using multiple LPARs for the workload.
- Dedicate memory to the LPAR based on the size of the environment that is planned. When over-committing memory in the hypervisor, help ensure that you have the proper auxiliary memory available (page in z/VM or swap in KVM) to avoid out of memory crashes.
- Dedicate or share core capacity according to the LPAR weight definition.
- Use at least two network cards for port aggregation and the Link Aggregation Control Protocol (LACP). The LACP requires that the ports are in promiscuous mode, and they must be dedicated to the partition.
- Consider using LinuxONE internal networking with HiperSockets or shared network cards.
- Use at least two I/O attachment cards (FICON or FCP).
- For virtualization in LPARs, consider z/VM or KVM and the VSWITCH topology.

2.9.2 Workload availability

To help ensure workload availability, consider using the following features:

- Failover and DR partitions for when your entire partition needs to reappear in another location.
- Volume access and crypto or key access.
- VM duplication: Keep VMs or containers available to run workload in an emergency. You will not avoid an outage, but a spare VM might shorten the Mean Time to Repair (MTTR).
- Dead guest relocation: The shutdown and startup might be faster than using guest mobility.

3

Deployment scenarios for resiliency

This chapter describes possible LinuxONE implementation scenarios, beginning with a single site and two IBM Storage Systems. Then, you expand the implementation to 2-site and 3- site scenarios. You explore each scenario's capabilities and possible variations in terms of their implemented resources.

This chapter contains the following topics:

- 3.1, "LinuxONE single-site scenarios" on page 62
- 3.2, "Multi-site scenario" on page 68
- 3.3, "Continuous availability scenario" on page 76

© Copyright IBM Corp. 2024.

3.1 LinuxONE single-site scenarios

This section describes the following LinuxONE single-site scenarios:

- Reliable base
- Flexible site

3.1.1 Reliable base

This section describes the following characteristics of the reliable base scenario:

- Reliable base scenario details
- Resiliency level of the reliable base scenario

Reliable base scenario details

This scenario is designed as a “starter” version of resiliency, with a focus on vertical scaling. It has the following components:

- One single physical site
- One LinuxONE system
- One Storage System
- z/VM as the hypervisor, and Extended Count Key Data (ECKD)[1] disks with HyperPAV
- One set of software

As shown in Figure 3-1, in this proposed scenario you have one IBM LinuxONE system running at a single physical site. The LinunxONE central processor complex (CPC) runs z/VM with multiple Linux guests to support the workload. LinuxONE CPC is configured to use Single System Image (SSI) and IBM Operations Manager for z/VM.

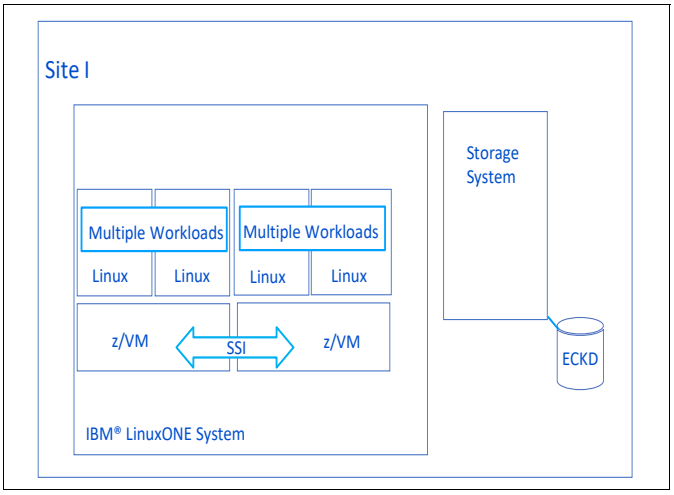


Figure 3-1 Reliable base implementation view

[1] In this chapter, ECKD and Count Key Data (CKD) are used interchangeably.

Resiliency level of the reliable base scenario

The workloads run on a single LinuxONE machine.

- Characteristics:
  - Storage hardware and LinuxONE hardware failures that are not addressed by the reliability, availability, and serviceability (RAS) capabilities are both single points of failure (SPOFs). Storage failover is not available.
  - All workloads do not run until a LinuxONE hardware failure is addressed.
  - Recovery tasks are performed manually, so the outage will likely extend to hours. There is inconsistent systems management.
- Benefits: Relies only on LinuxONE platform RAS.
- Improvements:
  - You can enhance Resiliency for a single IBM LinuxONE platform by using a clustering technology with data sharing across two or more Linux images.
  - Introduce an extra Storage System to extend access to data if there is a hardware failure.
  - Include Synchronous Copy with Peer-to-Peer Remote Copy (PPRC) and HyperSwap to improve recovery time and resiliency.
  - Develop and implement processes for handling problem and change management.
  - Review or plan application and processes upgrades to ensure that rapid failover can be accomplished.
  - Implement and use z/VM SSI and z/VM Operations Manager.

3.1.2 Flexible site

This section describes the following characteristics of the flexible site scenario:

- Flexible site scenario details
- z/VM Parallel Access Volume and HyperPAV
- Synchronous data replication
- More flexible site scenario considerations
- Network
- Resiliency level of the flexible site scenario

Flexible site scenario details

This scenario expands on the reliable base scenario by adding redundancy for compute, storage, and the network at the single site, which grants greater control of planned outages. It has the following components:

- One single physical site
- Two LinuxONE systems
- Two (or more) Storage Systems
- z/VM as the hypervisor, and ECKD disks with HyperPAV
- One set of software per system
- z/VM SSI

As shown in Figure 3-2, in this proposed scenario, there are two IBM LinuxONE systems running at a single physical site. The LinuxONE CPCs are running z/VM with multiple Linux guests to support the workload. LinuxONE CPCs are configured to use z/VM SSI and IBM Operations Manager for z/VM.

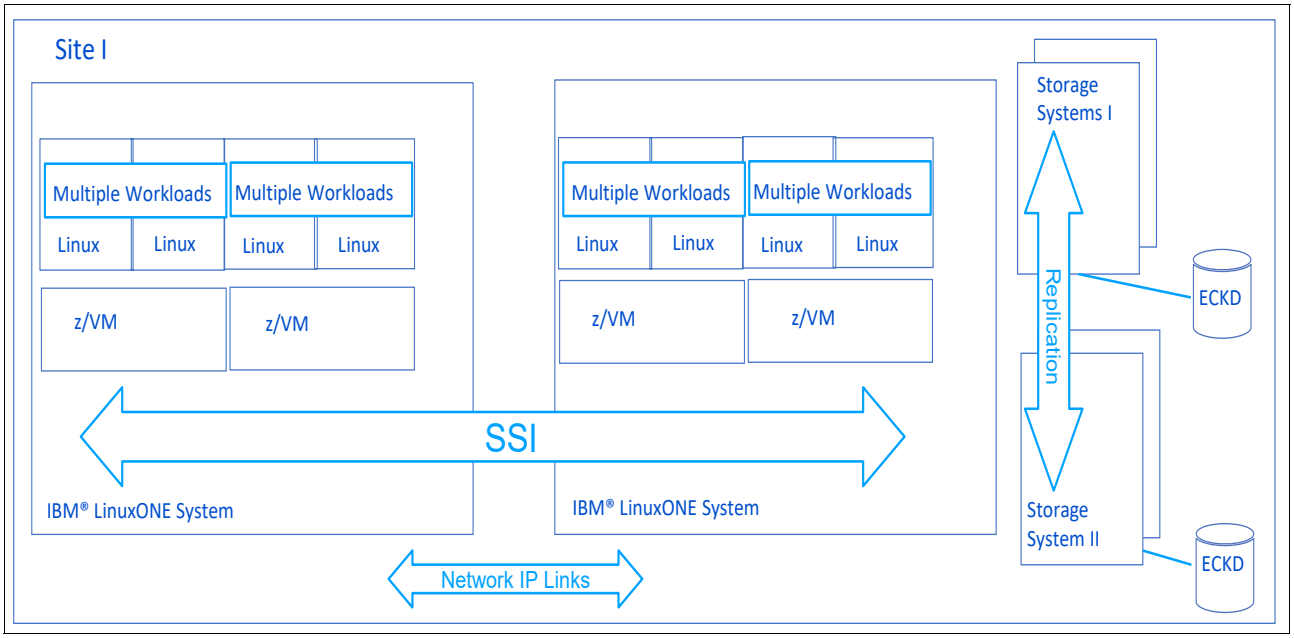


Figure 3-2 Flexible site implementation view

Running your workload on a single site constitutes a SPOF. Any disturbance in the electrical facilities or in any other infrastructure capabilities at the site causes an unplanned outage.

The benefits of this scenario are related to the fact that your data is fully replicated between the Storage Systems by using synchronous data replication (see “Synchronous data replication” on page 66) and z/VM SSI, which enables live migration of Linux guests among the four z/VM logical partitions (LPARs).

Also, two LinuxONE systems running the workload provides extended resiliency. If there is a hardware or software failure in one of the systems, the remaining active system can run the workload. The impact of a system failure depends on how the workloads are split between both systems, and on the capability of the remaining system to run them. A client can use Capacity on Demand (CoD) or Flexible Capacity for Cyber Resiliency capabilities to activate extra system capacity to run the workload in one system.

Figure 3-3 on page 65 shows an example with four z/VMs and one Linux on IBM Z LPARs that are defined, and three workload groups. One Kubernetes cluster group is formed by partitions 1, 2, and 3, 4 holds a web server, and partition 5 runs a database workload. The Storage Systems data is replicated with synchronous replication, which eliminates the storage SPOF. In this example, z/VM is running on LPARs I, II, III, and IV, and LPAR V is running Linux on IBM Z.

The LPAR configuration is “mirrored” on both LinuxONE systems to share the workload and facilitate the recovery process if there is a system hardware or software failure.

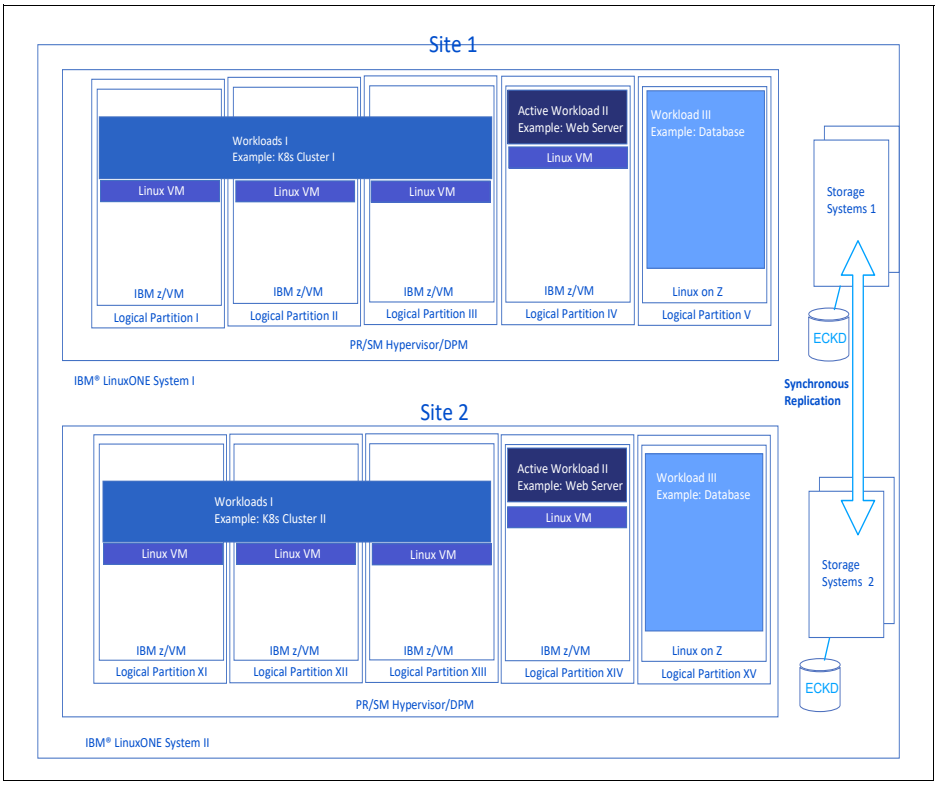


Figure 3-3 Flexible site reliable base detailed implementation example

z/VM Parallel Access Volume and HyperPAV

If the direct-access storage device (DASD) subsystem supports Parallel Access Volume (PAV) or HyperPAV, z/VM can use a real volume's alias devices to run more than one guest I/O operation at a time to the volume. When z/VM uses alias devices, the guests' I/O operations tend not to interleave or block each other. In turn, the guests' I/O operations tend to experience reduced real-volume I/O queuing and reduced response time. Reduced response time yields guest applications that run more quickly.

For performance reasons, in this scenario we opted to use HyperPAV technology, which enables concurrent I/O operations to a single disk volume.

Figure 3-4 shows how HyperPAV is leveraged by z/VM. Multiple I/O operations are initiated by the z/VM guests and handled by the z/VM I/O subsystems that are responsible for the selection of the base device address and the available aliases, which enable parallel simultaneous operations to the same volume to take place. The z/VM Control Program (CP) I/O Subsystem drives real concurrent Start SubChannel (SSCH) I/O operations to the alias devices.

Both the base and the aliases devices addresses must be in the same subchannel set (SS) to be recognized by z/VM. For more information, see Multiple Subchannel Set Support.

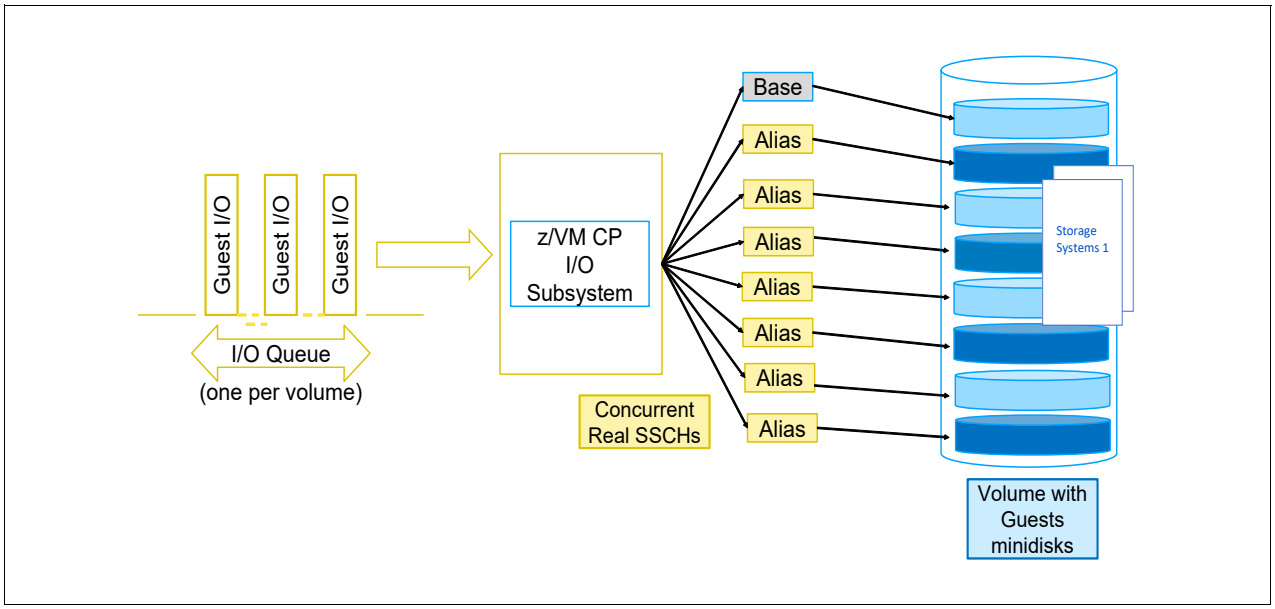


Figure 3-4 Multiple simultaneous disk operations by using z/VM HyperPAV

Synchronous data replication

Figure 3-2 on page 64 shows two Storage Systems that are configured to use PPRC. PPRC is a hardware solution that provides rapid and accurate disaster recovery (DR) and a solution to workload movement and device migration. Updates that are made on the primary DASD volumes are synchronously shadowed to the secondary DASD volumes. The local storage subsystem and the remote storage subsystem are connected through a communications link that is called a PPRC path. The protocol that is used to copy data by using PPRC is Fibre Channel Protocol (FCP). PPRC is configured in the Storage Systems by using local administration capabilities. In the case of IBM DS8000, one way of performing the configuration is by using its Hardware Management Console (HMC) or its CLI.

More flexible site scenario considerations

This scenario used one Linux on IBM Z partition and four z/VM Hypervisor partitions per LinuxONE system.

Using Kernel-based Virtual Machine instead of z/VM

If you Kernel-based Virtual Machine (KVM) with ECKD disks instead of z/VM, you can enable PAV or HyperPAV features in your Storage System. KVM assigns unique IDs to its DASDs and manages the alias devices. For more information about configuring, preparing, and working with DASDs, see *Device Drivers, Features, and Commands*, SC33-8411.

Using Fixed-Block Architecture[2] instead of ECKD

Fixed-Block Architecture (FBA) is an IBM term for the hard disk drive layout in which each addressable block on the disk has the same size (4-byte block numbers) and a new set of command codes.

In addition to supporting the emulation of count-key data (CKD) volumes, the DS8000 series Storage System (and other vendors' storage servers) support the definition of FBA volumes. These are known as logical unit numbers (LUNs) in a storage area network (SAN).

ECKD is a DASD data recording format that was introduced in 1964 with IBM System/360, and it is still emulated. It is a self-defining format with each data record represented by a Count Area that identifies the record and provides the number of bytes in an optional Key Area and an optional Data Area. ECKD blocks are usually addressed by CCHHR (CC=Cylinder; HH=Head <Track>; R = Record <Block>). This is in contrast to devices that use a fixed sector size or a separate format track.

Internally, all modern storage servers use FBA. For example, the DS8000 storage servers emulate CKD DASD volumes but the underlying technology is all FBA.

If you use KVM with FBA disks, then IBM Storage Scale for clustering can be an alternative. IBM Storage Scale, based on technology from IBM GPFS, is a high performance, shared-disk file management solution that provides fast, reliable access to data from multiple servers.

Applications can readily access files by using standard file system interfaces, and the same file can be accessed concurrently from multiple servers and protocols. IBM Storage Scale is designed to provide high availability (HA) through advanced clustering technologies, dynamic file system management, and data replication. IBM Storage Scale can continue to provide data access even when the cluster experiences storage or server malfunctions.

Network

The network capabilities of the LinuxONE platform are described in 2.5, “LinuxONE Virtual Network” on page 37.

Resiliency level of the flexible site scenario

The workload runs on a single site with multiple LinuxONE machines

- Characteristics:
  - All the workloads can be moved between systems if there is hardware maintenance or an outage.
  - Storage and network failover is not automated.
  - Manual recovery is used if there is an outage.
  - There is the risk of an entire data center outage.
- Benefits:
  - The data on disk is replicated between the two Storage Systems.
  - The hardware and software are not SPOFs.
- Improvements:
  - Introduce GDPS to improve automation if there is a failure or system outage.
  - Leverage HyperSwap to improve recovery time.
  - Improve processes for handling problem and change management.

[2] In this publication, the terms Fixed Block (FB) and Fixed-Block Assignment (FBA) are used interchangeably.

- Review, plan, and test applications and processes for changes and upgrades to ensure that a rapid failover can be accomplished.
- Implement and leverage z/VM SSI and z/VM Operations Manager.

3.2 Multi-site scenario

This scenario introduces one LinuxONE CPC and its associated storage into a second data center, which eliminates the physical site recovery point of failure. This step is necessary on the journey to high availability and disaster recovery (HADR).

3.2.1 Multi-site scenario details

In the multi-site scenario, the examples that follow use the following components:

- Two physical sites at metro distance (up to 300 km with DS8870 Metro Mirror (MM))
- One or more LinuxONE systems per site
- Two or more Storage Systems per site with synchronous replication:
  - Local or between sites replication
  - z/VM as hypervisor, and ECKD disks on both sites with HyperPAV

The multi-site resiliency scenario is basically the flexible site scenario with components that are replicated to a secondary site.

Figure 3-5 on page 69 shows the 2-site basic components and Storage Systems data that is replicated synchronously between the sites.

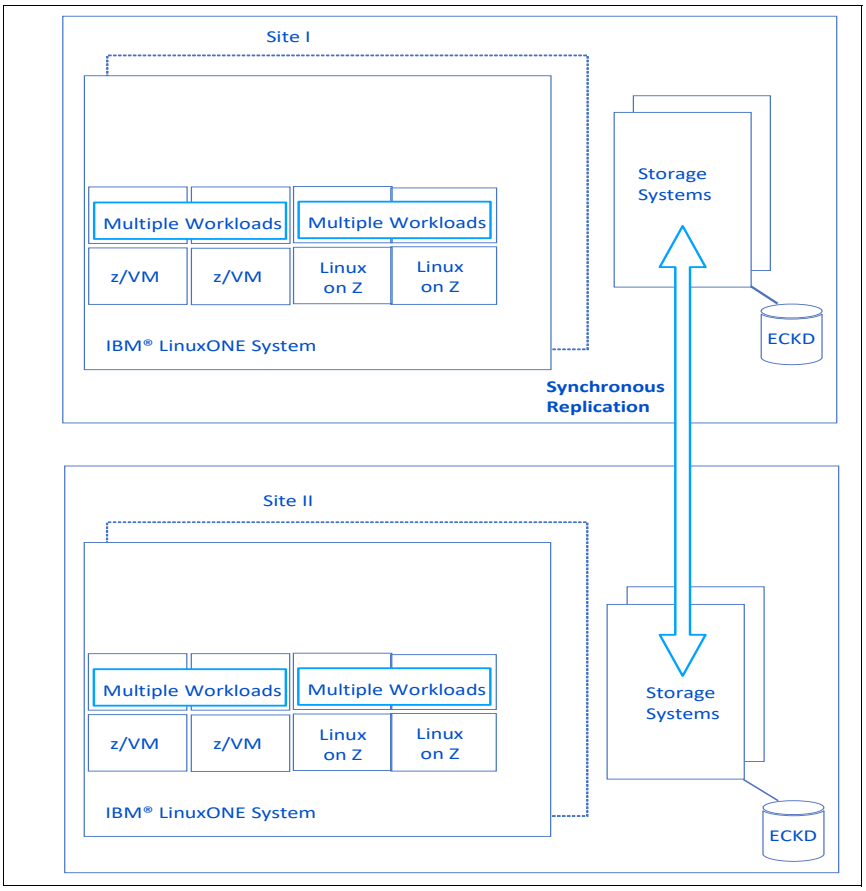


Figure 3-5 Multi-site resilient scenario components

3.2.2 Applicable GDPS capabilities for a multi-site scenario

This section describes the following topics:

- GDPS Virtual Appliance
- GDPS Metro
- GDPS Extended Disaster Recovery

GDPS Virtual Appliance

GDPS Virtual Appliance is exclusive to IBM LinuxONE. It includes GDPS Metro and GDPS Extended Disaster Recovery (xDR) capabilities, which are used in Hot-Warm and Hot-Hot configurations. For more information, see “GDPS Virtual Appliance” on page 53.

GDPS Metro

GDPS Metro can manage the Multi-Target Metro Mirror configuration, which extends PPRC management and HyperSwap capabilities to support two synchronous MM relationships from a single primary volume. Each leg is tracked and managed independently. This provides extra data protection if there is a disk subsystem failure or local disaster scenario. When you use an ECKD formatted disk, GDPS Metro can provide the reconfiguration capabilities for z/VM and its guests and Linux on IBM Z servers and data. To support planned and unplanned outages these functions are extended to KVM on LinuxONE starting with GDPS 4.1.

GDPS Metro provides the following recovery actions:

- “Restart in place” of failing operating system (OS) images.
- z/VM Live Guest Relocation (LGR) management.
- Manage z/VM LPARs and z/VM guests, which include Linux on IBM Z.
- Heartbeat checking of Linux guests.
- Disk error detection.
- Data consistency with freeze functions across Linux.
- Site takeover or failover of a complete production site.
- Single point of control to manage disk mirroring configurations.
- Coordinated recovery for planned and unplanned events.

More support is available for Linux running as a guest under z/VM:

- “Restart in place” of failing OS images.
- Ordered Linux node or cluster startup and shutdown.
- Coordinated Planned HyperSwap (PHS) and Unplanned HyperSwap (UHS) of disk subsystems, which are transparent to the OS images and applications that use the disks.
- Transparent disk maintenance and failure recovery with HyperSwap across Linux applications.

GDPS Extended Disaster Recovery

GDPS Metro provides management extensions for heterogeneous platforms (GDPS xDR) to fully manage either z/VM systems and their guests, KVM and its guests, and Linux running on LinuxONE environments, which provide full, end-to-end support, which includes disk management with freeze and PHS and UHS support, systems management and monitoring. This support also applies to z/VM and KVM on FBA formatted disk. For more information, see “Extended Disaster Recovery for z/VM with GDPS Virtual Appliance” on page 55.

3.2.3 Multi-site possible configurations

Because LinuxONE CPCs and the Storage System servers are at two different physical sites, reliability and availability are improved compared to the previous scenarios. This is also a basic required scenario for DR.

There are three configuration possibilities for setting up 2-site scenarios:

- Hot-Cold (Active-Passive): See “Characteristics of a Hot-Cold (Active-Passive) deployment” on page 57.
- Hot-Warm (Active-Idle): See “Characteristics of a Hot-Warm (Active-Idle) deployment” on page 58.
- Hot-Hot (Active-Active): See “Characteristics of a Hot-Hot (Active-Active) deployment” on page 58.

Hot-Cold (Active-Passive) configuration

In a Hot-Cold configuration, the secondary (DR) site can be a “dark” or “dormant” site that is ready to be used if the primary site fails. Data is replicated between the sites.

The workload is not split, and each site is configured to handle all operations. Manual intervention is normally required to activate extra resources and the partitions at the cold (passive site). Therefore, the cold environment, which is often used in a DR situation, needs more time to activate.

The LinuxONE CPC on the secondary site might have a temporary capacity record that is installed, such as On/Off Capacity on Demand (OOCoD), Capacity BackUp (CBU), or Flexible Capacity for Cyber Resiliency, which is ready to be activated manually by the operations staff. For more information, see “Capacity on Demand (temporary upgrades)” on page 22.

The objective of these temporary capacity records is to bring the secondary site capacity to a level that enables the partitions and workloads that were previously running on the primary site to run on the secondary site without any impact to the user or overall system performance. For more information about temporary capacity, see the Capacity on Demand User’s Guide.

Figure 3-6 shows a more elaborate example of an active-passive, 2-site implementation. In this scenario, the LPARS on Site2 are defined but not activated because the total resources that are required by the defined LPARs, which are the same as the resources that are used by the primary site, might not be available until a temporary capacity record is activated. Activating a temporary capacity record in this scenario requires manual intervention by using the LinuxONE HMC. Because of these manual interventions characteristics, the recovery time objective (RTO) can increase.

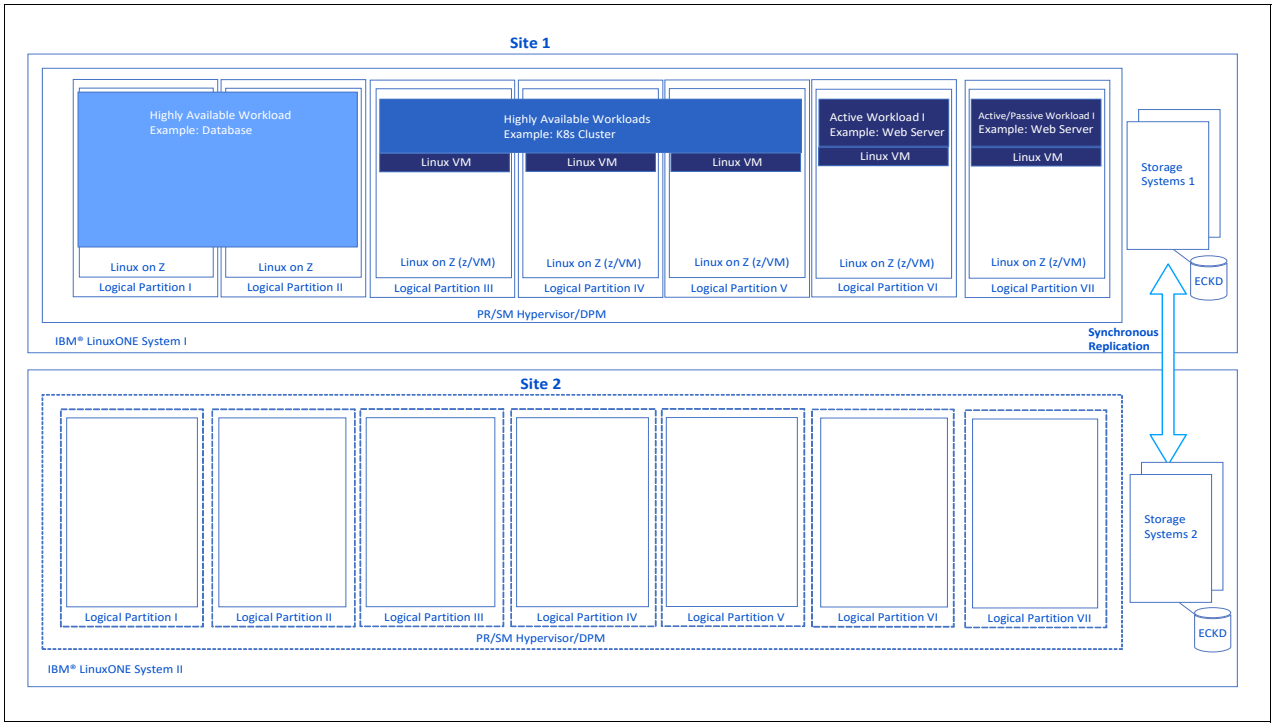


Figure 3-6 Hot-Cold (Active-Passive) 2-site implementation example

Hot-Warm (Active-Idle) configuration

In a Hot-Warm (Active-Idle) environment, Site1 runs the entire production workload, and Site2 (the DR data center) has a system that is on and has an active basic setup, such as hypervisors, and support for the replication of data or special software for automatic failover, such as IBM GDPS Virtual Appliance. Site2 has all the necessary resources to run the Site1 workload. However, all resources are in an “idle” state. Normally, the Storage System servers at Site1 and Site2 are set up for the replication and mirroring of all data from production.

If there is a Site1 failure, Site2 must be activated manually through GDPS Virtual Appliance automation or by using user-defined scripts.

Figure 3-7 shows an implementation of a Hot-Warm (Active-Idle) environment.

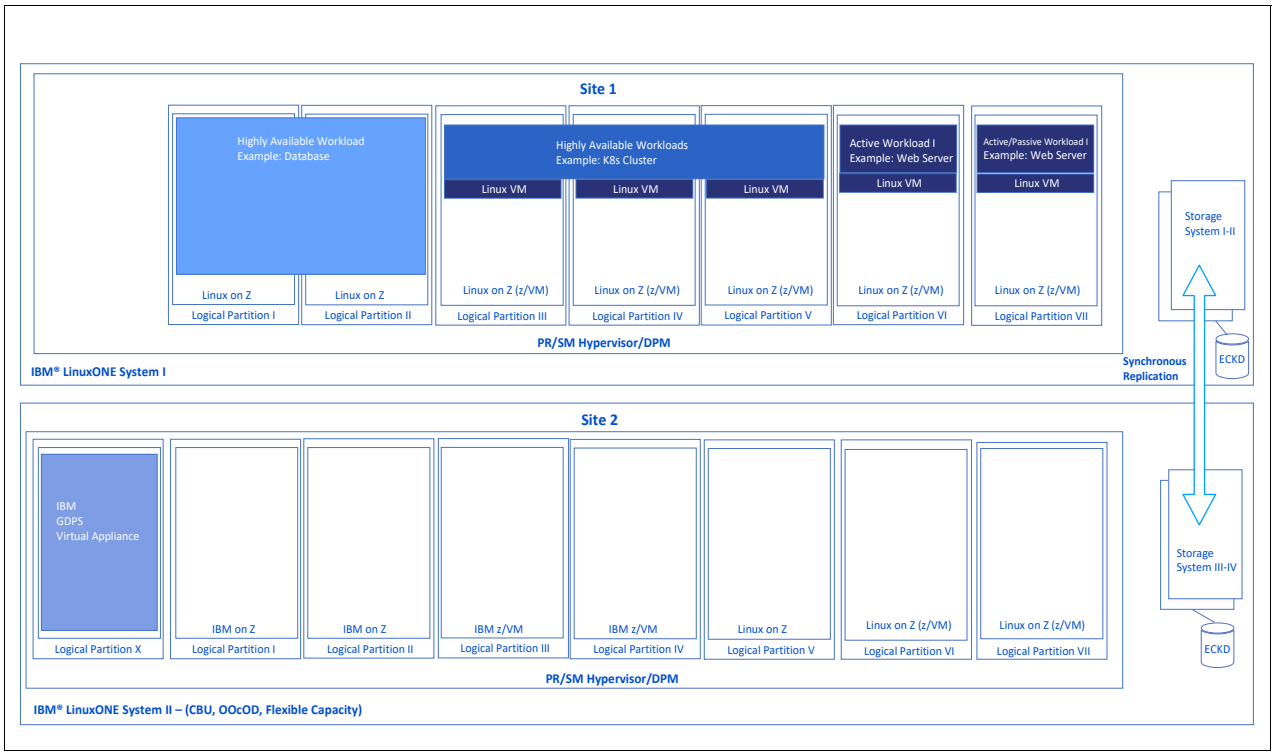


Figure 3-7 Hot-Warm (Active-Idle) 2-site implementation example

The GDPS Virtual Appliance is a software appliance, which means that it is a self-contained, OS, application, middleware, API, and GUI, which is all in one IBM LinuxONE LPAR software image. The GDPS Virtual Appliance can start and stop z/VM or KVM hypervisors and all Linux guests that are running on these hypervisors. The GDPS Virtual Appliance also manages disk replication direction between your sites, as shown in Figure 3-9 on page 74. The “sites” that are shown can be data centers that are separated by up to 124 miles (200 km), or different racks that are separated only by a few feet in the same data center. For DR, the distance between sites is important because it can save your business if there is a city-wide or region-wide disaster.

Hot-Hot (Active-Active) configuration

In terms of resiliency, the most effective solution is where the two sites are active concurrently and they are configured to take over the entire production environment workload whenever one of the sites fails or a disaster strikes. During normal operation, excess capacity at each site is consumed by lower priority work like development or test activities. Also, each site is configured with sufficient capacity to handle normal operations. In a failover situation, extra resources can be enabled and the entire workload runs on a single site.

Figure 3-8 on page 73 shows a more detailed 2-site (Active-Active) implementation. The two storage controllers are synchronously replicated, so the sites may become the production site without compromising the information that is written (and mirrored) to the disks in both controllers if one of the sites goes down. In a normal situation (both sites up), the workload can be shared between the CPCs on both sites.

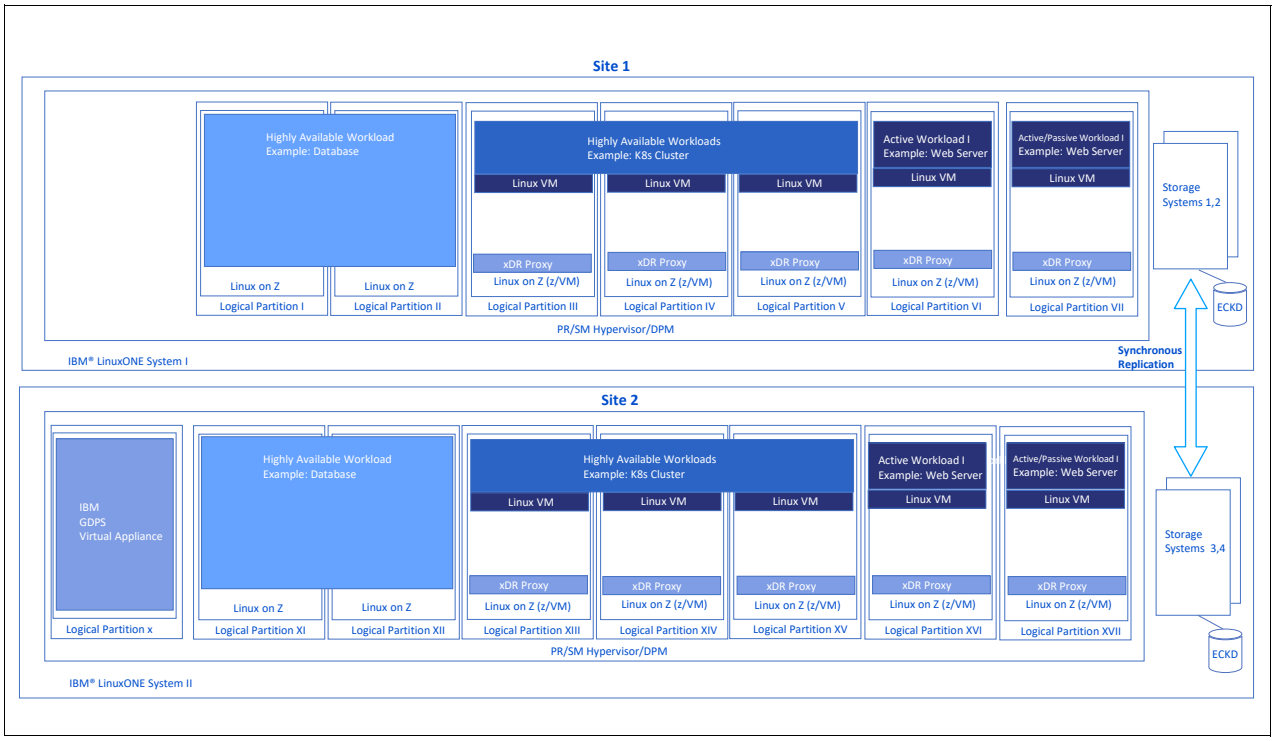


Figure 3-8 Two sites in an Active-Active configuration implementation example

In addition to showing the workloads that are running on both sites, this example has an extra partition (LPAR XII) that runs GDPS Virtual Appliance. For more information about GDPS Virtual Appliance, see “GDPS Virtual Appliance” on page 53. On each z/VM image in Site1, there is an extra xDR proxy LPAR. The xDR proxies are guests that are dedicated to provide communication and coordination between the z/VM and GDPS Virtual Appliance.

With active/active configurations, you might have a router and a load-balancer in front of the cluster to balance and split the incoming requests among the active nodes in the cluster. When a system fails, its service workload is migrated to an active node. When one active member fails, the resources are still running on the other active members, and the new incoming service is uninterrupted. Systems must have sufficient hardware resources to handle extra work if there is an outage of one system or work must be prioritized and service restricted if there is a system failure.

Also, based on Figure 3-9, a GDPS Virtual Appliance partition should be placed at the secondary (DR) site. The xDR Proxies partitions are on both sites and each z/VM partition.

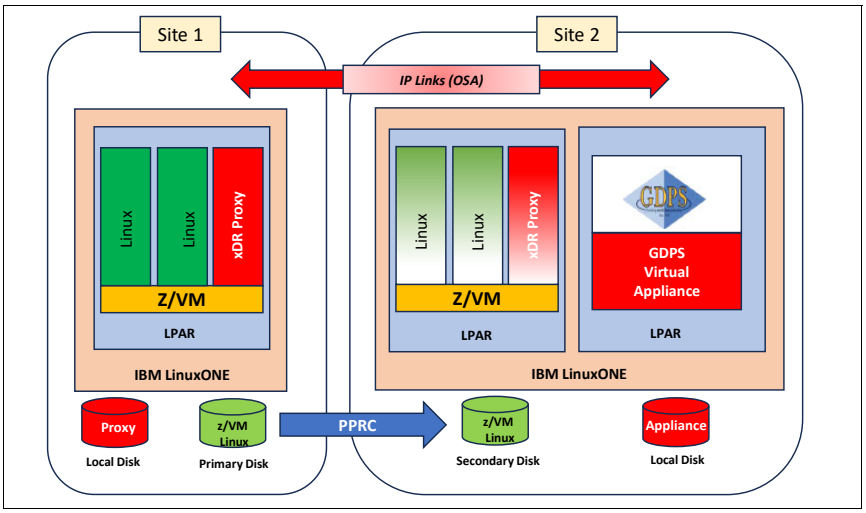


Figure 3-9 GDPS Virtual Appliance basic implementation diagram

With two sites, GDPS Virtual Appliance and GDPS Metro MultiPlatform Resiliency for IBM Z (GDPS xDR) can be configured to automate the process of transferring the workloads; swapping Storage Subsystems disk access; activating temporary records; and restarting hypervisors such as z/VM and z/KVM and their respective guests to keep production running if there is an unexpected site unexpected, disaster, or planned or unplanned outage.

3.2.4 Implementing z/VM Single System Image

z/VM SSI is described in “z/VM Single System Image” on page 31. SSI enhances the z/VM systems management, communications, disk management, device mapping, virtual machine (VM) definition management, installation, and service functions to enable multiple z/VM systems to share and coordinate resources within a SSI structure. This combination of enhanced functions provides the foundation that enables Live Guest Relocations (LGR), which is the ability for a Linux guest to be moved from one z/VM system to another one within the SSI cluster.

z/VM SSI is applicable to both scenarios that were presented so far. It applies to a single site with one or more CPCs (Scenario 1), and to a dual site (Scenario 2) at metro distance. In a single site or dual site, SSI requires communication between the z/VM images. That communication is normally provided by channel-to-channel (CTC) connections.

3.2.5 HyperSwap function

HyperSwap, which is available with the GDPS Virtual Appliance, can transparently swap disks between two sites. The power of automation enables you to test and perfect the actions to take, either for planned or unplanned changes, which minimize or eliminates the risk of human error.

HyperSwap is one of the offerings in the GDPS family along with GDPS Metro that offers the potential of zero data loss, and that can achieve the shortest RTO, which typically less than 1 hour after a complete site failure.

HyperSwap is also one of the only members of the GDPS family along with GDPS Metro that is based on hardware replication and can manage the production LPARs.

3.2.6 PAV, HyperPAV, and HyperSwap in z/VM

In a z/VM environment, you can use HyperPAV *and* HyperSwap but then HyperPAV must be attached to a system and not dedicated to a guest, Therefore, you can use any guest as an OS that is not configured to control the features of HyperPAV or has no knowledge of the HyperPAV architecture.

There is a tradeoff between overall DASD I/O subsystems performance and the ability to perform a quick, dynamic, and automated recovery process for a Storage System failure:

- PAV and HyperPAV improve performance by enabling multiple I/Os to be performed concurrently against one disk volume.
- HyperSwap improves recovery time if there is a failure in the data replication process or in a Storage System hardware incident.

With z/VM, the usage of virtual alias devices (PAV or HyperPAV) in a guest, whether dedicated or virtual, is considered an unsupported configuration for HyperSwap.

3.2.7 Resiliency level of multi-site scenarios

The workloads run on two (or more) LinuxONE machines at different sites.

- Characteristics:
  - Requires a higher investment in equipment and infrastructure.
  - Active-active requires that machines at both sites have extra capacity or temporary capacity that are available for activation so that they can support all workloads.
  - Leverages GDPS Virtual Appliance, GDPS xDR, GDPS Metro, and HyperSwap.
- Benefits:
  - Provides near-continuous availability (CA) and continuous operations (CO) (active-active).
  - Minimal impact due to system and data outages (redundancy in software, hardware, and data).
  - Full data replication between sites and Storage Systems.
  - Use the GDPS Virtual Appliance GUI to configure and monitor the environment without requiring extra operations skills.
  - With GDPS Virtual Appliance and the selected scenario implementation (active-cold, active-warm, or active-active), the failover process can be automated by using service management processes.
  - In a failover scenario, the impact can be minutes to hours depending on the level of detection and automation that is implemented.
- Improvements:
  - Proper workload distribution and routing between sites and resources (active-active).
  - Test and exercise all possible failure DR situations.
  - Determine the levels and frequency of maintenance of all installed equipment.
  - Improve processes for handling change and automated problem management.

- Review or plan application updates to ensure that rapid failover can be accomplished.
- CA.

3.3 Continuous availability scenario

The section describes the following topics:

- Continuous availability scenario details
- Continuous availability scenario environment
- GDPS Continuous Availability capabilities
- Resiliency level of continuous availability
- “Eight nines” (99.999999%) availability

3.3.1 Continuous availability scenario details

This scenario has the following components:

- Multiple sites at metro and global distances.
- One or more LinuxONE systems per site.
- Two or more Storage Systems with synchronous and asynchronous replication, with local or emote replication between sites.
- z/VM as the hypervisor and CKD disks at all sites with HyperPAV.
- KVM as the hypervisor and CKD disks at all sites.
- Linux in an LPAR with CKD disks.
- A z/OS partition, which is required for Global Mirror (GM) implementation.
- GDPS xDR.

3.3.2 Continuous availability scenario environment

The environment for this implementation includes current IBM Z clients running a z/OS Parallel Sysplex environment that uses GDPS capabilities and extends these capabilities to their workloads that are not running on their current IBM Z platform. Placing the distributed workloads on the IBM Z platform is possible, and depending on the GDPS licenses that the clients already own, they might need to add only GDPS xDR - Multiplatform Resiliency for their IBM Z systems GDPS licenses.

Figure 3-10 on page 77 shows an environment where IBM Z is running z/OS in an IBM Parallel Sysplex® environment along with distributed Linux in an LPAR, and native Linux, z/VM, and KVM Linux guests running on LinuxONE servers. LinuxONE servers do not support z/OS.

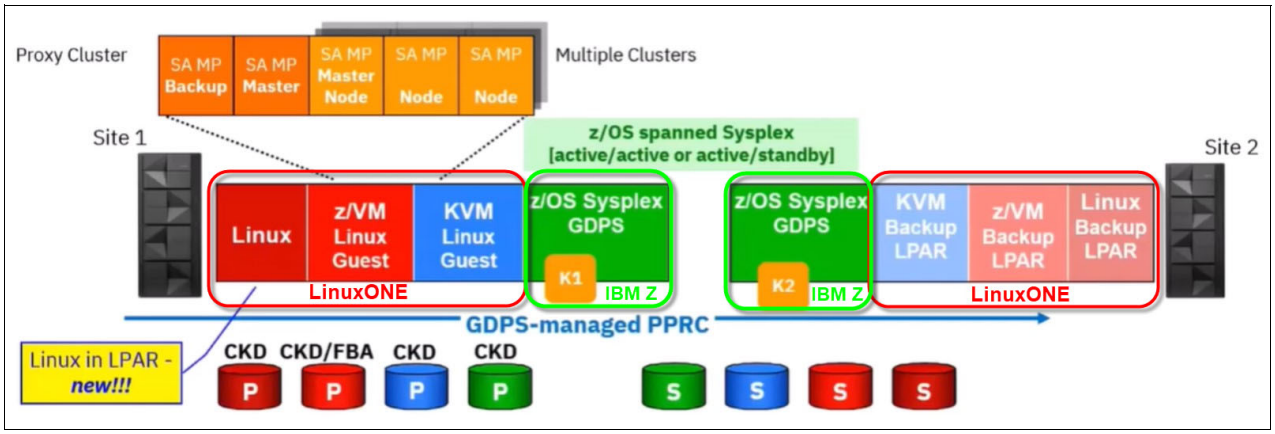


Figure 3-10 Example of two IBM Z systems running z/OS Sysplex, Linux on IBM Z, KVM, and z/VM

3.3.3 GDPS Continuous Availability capabilities

The section describes the following topics:

- GDPS Metro
- GDPS 3-site solution
- GDPS Metro, Global Mirror, and Metro Global Mirror
- Fixed Block disk management

GDPS Metro

GDPS Metro can manage the Multi-Target Metro Mirror configuration, which extends PPRC management and HyperSwap capabilities to support two synchronous MM relationships from a single primary volume. HyperSwap supports ECKD and xDR managed FB disks. Each leg is tracked and managed independently, which provides extra data protection if there is a disk subsystem failure or local disaster scenario.

GDPS Metro provides management extensions for heterogeneous platforms (xDR) to fully manage either z/VM systems and their guests, Linux in LPARs, and KVM on LinuxONE environments to provide full end-to-end support, which includes disk management with freeze and PHS and UHS support, systems management, and monitoring. This support also applies to z/VM and KVM on FBA formatted disks.

As shown in Figure 3-10, GDPS Global requires two z/OS systems images (K1 and K2) to manage, monitor, and run recovery scripts.

GDPS 3-site solution

GPDS also supports 3-site configuration solutions by using a combination of MM and GM solutions. A 3-site solution can combine the advantages of metro-distance BC and regional or long-distance DR.

GDPS Metro, Global Mirror, and Metro Global Mirror

You can use GDPS Metro with GDPS GM to manage the configuration across all formats of data because GM is not limited to IBM Z formatted data.

A typical configuration has the secondary disk from an MM Remote Copy configuration, which in turn becomes the primary disk for a GM Remote Copy pair. Data is replicated in a “cascading” fashion.

Metro Global Mirror (MGM) is a method of continuous, remote data replication that operates between three sites that are varying distances apart. MGM combines MM synchronous copy and GM asynchronous copy into a single session, where the MM target is the GM source. Using MGM and MGM with HyperSwap, your data exists on a second site that is less than 300 km away, and a third site that is more than 300 km away. MGM uses both MM and GM failover and failback to switch the direction of the data flow. This ability enables you to run your business from the secondary or tertiary sites.

Fixed Block disk management

GDPS Metro, and GDPS GM and MGM technology are extended to manage a heterogeneous environment of IBM Z and distributed systems LUNs, also known as Fixed-Block (FB) disk. If installations share their disk subsystems between the IBM Z and distributed systems platforms, GDPS Metro, GDPS GM, and GDPS MGM can manage the MM and GM remote copy configurations, and FlashCopy for distributed systems storage.

GDPS MM and GDPS GM are also designed to provide the following features:

- A single point of control and management for both FB and ECKD disk replication.
- A freeze capability to protect the consistency group for synchronous replication.
- Data consistency across IBM Z and distributed environments for applications that span multiple tiers.
- HyperSwap capability for FB devices with native Linux on IBM Z.

> **Note:** If the client has zFBA in a shared ECKD Consistency Group and GDPS does a HyperSwap, GDPS resets all non-HyperSwap capable systems (“inhibited”, Linux on IBM Z, KVM, and SSC) if there is an UHS. A Planned HyperSwap (PHS) sends an alert about those systems running, so a PHS is not performed until all such systems are down.

- GDPS Metro uses the DS8000 zFBA support.

Using FB disk management support enables GDPS to be a single point of control to manage business resiliency across multiple tiers in the infrastructure, which improves cross-platform system management and business processes.

Figure 3-11 on page 79 shows a possible configuration of an MGM environment. It is divided into a primary production site (Site1), a secondary production site (Site2), and a remote site (Site3). Between the primary and secondary production sites (Site1 and Site2), MM relationships can be configured in either direction. At both production sites, a GM is configured to the remote site. All connections are set up with redundant inter-switch links between fiber directors or switches, which are not shown in the figure.

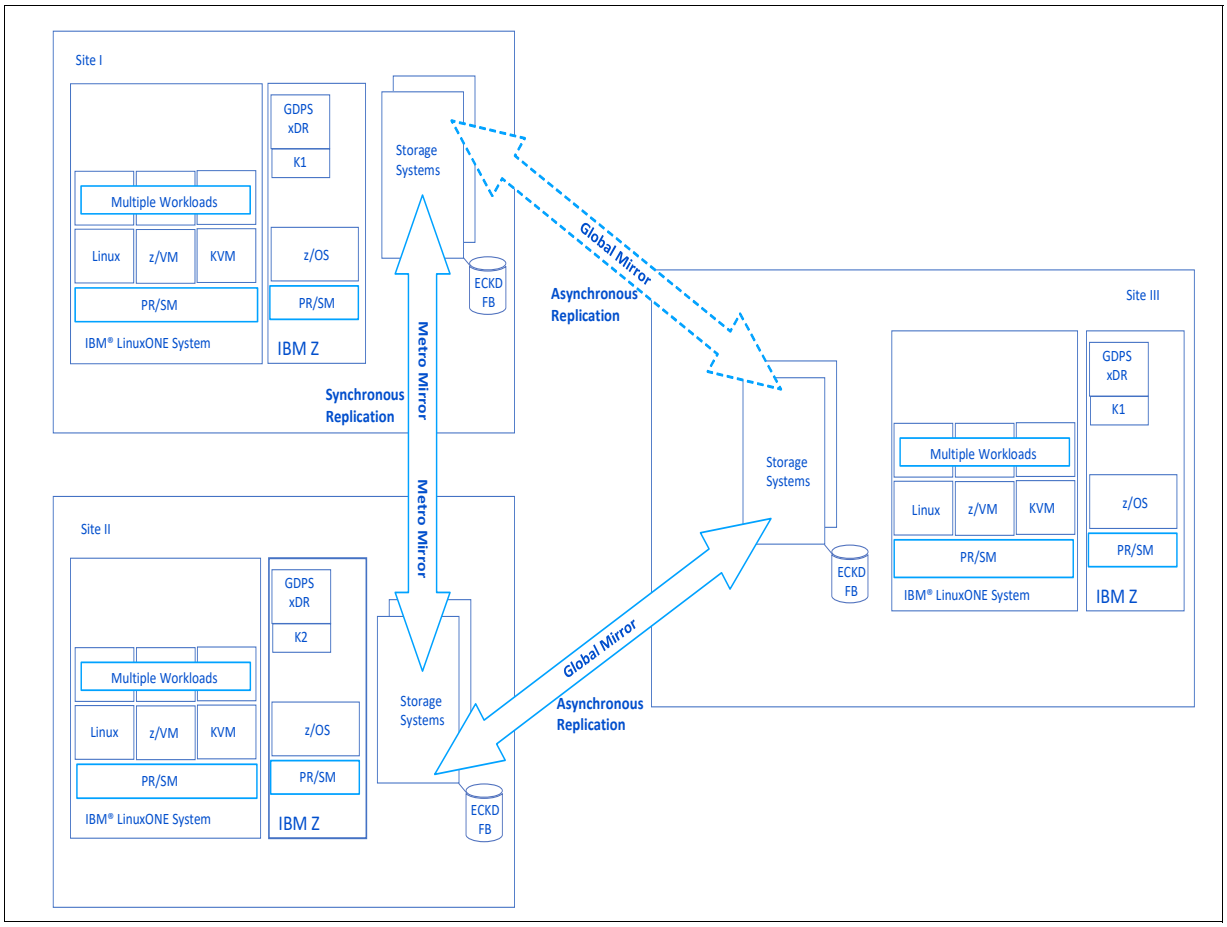


Figure 3-11 Multi-site fault-tolerant scenario with continuous availability example

To support this complex configuration, two z/OS partitions are required. Because LinuxONE cannot run z/OS, another IBM Z system is required at each site. The IBM Z systems hold the z/OS LPARs, GDPS xDR, and the K1 and K2 controlling systems.

The OSs must run on servers that are connected to the same Hardware Management Console (HMC) local area network (LAN) as the GDPS control system.

3.3.4 Resiliency level of continuous availability

- Characteristics:
  - Protects against local and regional planned or unplanned outages.
  - Provides CA of heterogeneous platforms with GDPS xDR.
  - Effective and efficient systems management.
  - GDPS automated takeover (site switch, HyperSwap, and restart).
  - GDPS proxies on z/VM (multiple) and backing by a K-sys in a Parallel Sysplex.
- Benefits:
  - Rapid restart.
  - Multiple LPARs cloned.
  - Data sharing (GDPS MGM).
  - GDPS full automation.
  - Provisioning spare capacity that is automatically activated by GDPS.

- Improvements:
  - GDPS and SafeGuarded Copy for snapshots of system contents.
  - Restoration of previous copies of workload for forensic investigation.

3.3.5 “Eight nines” (99.999999%) availability

IBM LinuxONE Emperor 4 systems with GDPS, and IBM DS8000 series with HyperSwap and running a Red Hat OpenShift Container Platform environment are designed to deliver 99.999999% availability.

Chapter 4, “Red Hat OpenShift Container Platform resiliency” on page 81 explores a configuration that is designed to allow this extended availability.

Disclaimer: IBM internal data that is based on measurements and projections was used in calculating the expected value. Necessary components include IBM LinuxONE Emperor 4; IBM z/VM 7.3 systems that are collected in an SSI, with each one running Red Hat OpenShift Container Platform 4.14 or later; IBM Operations Manager; GDPS 4.6 for management of data recovery and VM recovery across metro distance systems and storage, including Metro multi-site workload and GDPS Global; and IBM DS8000 series storage with IBM HyperSwap. A MongoDB 4.2 workload was used. Necessary resiliency technology must be enabled, which includes z/VM SSI clustering, GDPS xDR Proxy for z/VM, and Red Hat OpenShift Data Foundation 4.14 for management of local storage devices. Application-induced outages are not included in these measurements. Other configurations (hardware or software) might provide different availability characteristics.

4

Red Hat OpenShift Container Platform resiliency

This chapter describes Red Hat OpenShift Container Platform resiliency when it is running on IBM z16 and IBM LinuxONE 4 servers.

This chapter also described the means of availability calculation, which is an IBM differentiation function for the Red Hat OpenShift Container Platform workload; the required resiliency configuration; potential points of planned and unplanned outage; technical mitigations; and service interruption, disaster recovery (DR), and cyber resiliency.

Red Hat OpenShift Container Platform provides continuous operation and service resiliency when running on IBM LinuxONE. By combining the Red Hat Open Container Platform clustering technology with enterprise-level compute (IBM LinuxONE) and storage (IBM DS8000) technology with an appropriate virtual infrastructure (IBM PR/SM, IBM z/VM and associated automation, or IBM GDPS), a client workload can achieve over 99.999999% (eight nines) of uptime despite planned or unplanned outages.

This uptime can be achieved for data and enterprise services within the bounds of this configuration.

This chapter contains the following topics:

- 4.1, “Red Hat OpenShift Container Platform resiliency” on page 82
- 4.2, “Availability and resiliency for Linux workloads” on page 84
- 4.3, “Framework analysis” on page 89
- 4.4, “Disaster recovery” on page 106
- 4.5, “Resources” on page 109

© Copyright IBM Corp. 2024.

4.1 Red Hat OpenShift Container Platform resiliency

This section describes the following topics:

- Calculating availability
- Differentiators

4.1.1 Calculating availability

Availability is calculated based on service metrics, environment testing, and solution modeling. With compute redundancy, the availability ceiling is access to data on storage.

- Physical sites exist at 10^5 availability (99.999% uptime). As a best practice, use at least two sites.
- The implementation requires redundancy and automation for compute resources (IBM z16 and IBM LinuxONE, IBM z/VM, IBM GDPS, and Red Hat OpenShift Container Platform).
- DS8000 storage volumes have 10^6 availability (99.9999% uptime) on an individual basis.
- When DS8000 is managed by GDPS, you can reach at least 10^8 availability (99.999999% uptime).

Assumptions

Availability is based on an optimum configuration for the calculation of availability for Red Hat OpenShift Container Platform workloads that are running on LinuxONE hardware:

- Assume a configuration where all components (hardware and software) have appropriate backup and redundancy to allow for outage mitigation and help ensure minimal performance impact within client requirements:
  - Includes available space on the systems (CPU, storage, and memory) if there are planned outages and workload redistribution.
  - Includes configuring z/VM and Red Hat OpenShift Container Platform for high availability (HA) and redundancy (virtual resources are resources).
- Assume that the greater data center environment is configured for redundancy and HA:
  - Includes services such as DNS and load-balancing for applications that are running in Red Hat OpenShift Container Platform.
  - Includes physical cabling for networking.
  - Includes power and electricity, fire suppression, and other items.
- Assume a competent and non-malicious set of system administrators that test fixes on a development system that is isolated from production workloads.
- Assume a correctly configured z/OS system that is running GDPS Extended Disaster Recovery (xDR) in the enterprise. For more information, see the following resources:
  - *System z Parallel Sysplex Best Practices*, SG24-7817.
  - *IBM GDPS: An Introduction to Concepts and Capabilities*, SG24-6374.
- Assume that all pertinent security guidelines are followed regarding configuration of the included implementation.

Threats

The resiliency model is designed to withstand the following types of threats to service availability:

- Natural disasters

  A multi-site configuration should protect against natural disasters, but it on the size and scope of the disaster.

- Planned outages to systems
- Power outages

  Backup generators and power failover should already be a physical availability consideration.

- Replication failures (non-malicious or non-user-error)
  - Restore previous copies of a workload, which depends on forensic investigation.
  - Perform a data integrity check after, for example, a system upgrade.

Although security threats (cyberattack, malicious users, and ransomware) are acknowledged potential threats, they are outside the scope of this particular calculation. For more information about IBM LinuxONE Security and cyber resiliency, see The Value of LinuxONE Virtualization Security.

4.1.2 Differentiators

Here are the differentiators of this solution:

- IBM LinuxONE hardware

  Designed for maximal resiliency. Its reliability, availability, and serviceability (RAS) characteristics include core sparing; Redundant Array of Independent Memory (RAIM) and critical parts redundancy; for example, power supplies; cooling components; and a service network. LinuxONE provides vertical scalability for systems (more servers that are hosted with less hardware that can fail). At certain performance points, processor cores and memory can be added concurrently and transaction security can be achieved through on-chip symmetric encryption or shared Crypto Express domains.

- IBM Storage hardware (DS8000)

  Built for HA. With Extended Count Key Data (ECKD), it has more data integrity, redundancy, and availability compared to SCSI devices.

- IBM virtualization

  PR/SM and z/VM provide secure and scalable workload hosting that to help avoid planned outages.

- IBM GDPS

  Used in support of data replication, mirroring, and HyperSwap of disk volumes.

4.2 Availability and resiliency for Linux workloads

This section describes optimum configurations for the calculation of availability and resiliency for Linux based workloads on IBM LinuxONE hardware.

The workload under consideration is containerized applications (Red Hat OpenShift Container Platform). Database workloads, for example, Oracle, have extra or alternative considerations. The included calculations work within the boundaries of virtual infrastructure and containerization, but do not include individual applications. Although with this implementation Red Hat OpenShift Container Platform container workloads should be distributed, with load-balancing and redundancy for each, the resiliency of an application itself is not evaluated.

No claims are made specifically about zCX and z/OS hosted workloads in this document.

This guidance is not a client mandate; clients may optimize their environments to meet their needs, including modifications to achieve even higher redundancy.

4.2.1 Components of LinuxONE resiliency

The components of this approach are shown in Figure 4-1.

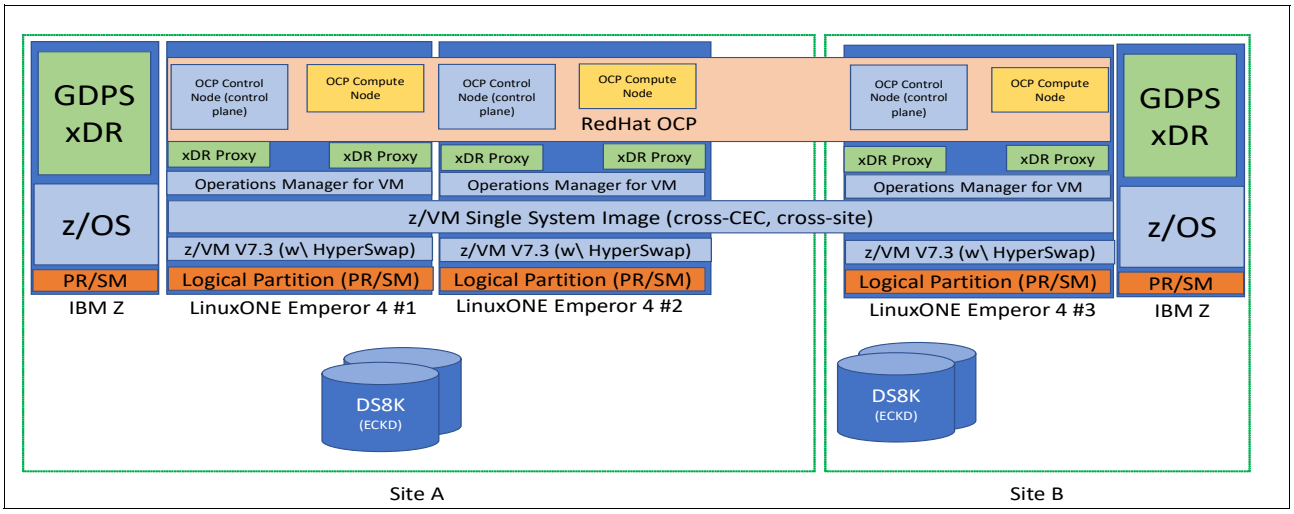


Figure 4-1 Structural diagram

- Two physical sites at a regional distance
- At least two Storage units (DS8000-series configured for ECKD
- One IBM z16 machine per site for running z/OS 2.5 and GDPS 4.5 (K-sys and xDR)
- LinuxONE machines with two or more central processor complexes (CPCs) (at least one per site) that are configured in PR/SM mode (not Dynamic Partition Manager (DPM))
- z/VM V7.3 with virtual networking, HyperSwap, and Single System Image (SSI):
  - Minimum of three z/VM partitions (out of eight maximum) in an SSI cluster.
  - z/VM 7.3 GA 09/2022 increases the SSI cluster to eight members.[1]

[1] The previous version of z/VM supported a maximum of four members in an SSI.

- IBM Operations Manager for z/VM 1.6.
- GDPS 4.5 xDR Proxies for z/VM.

- Red Hat OpenShift Container Platform 4.10 (CoreOS) that are configured for containerization resiliency:
  - Red Hat OpenShift Data Foundation 4.10 for local storage.
  - System Automation for Multiplatforms (SA MP) for Red Hat OpenShift Container Platform Control Node and Compute Node.
  - Control and compute nodes should not be converged in a single guest.

Here are some considerations about the environment that is shown in Figure 4-1 on page 84:

- Versions:
  - Red Hat OpenShift Data Foundation carries the same version number as Red Hat OpenShift Container Platform.
  - Operations Manager 1.6 supports earlier releases of z/VM.
  - Red Hat OpenShift Container Platform 4.10 runs on IBM z13®, IBM z14®, IBM z15®, and IBM z16. Version 4.9 is optimized to run on IBM z13, IBM z14, and z15, but it is not supported to run on the IBM z16.
- Hardware

  IBM Z on z16 and LinuxONE on LinuxONE IV machines must be configured to run in PR/SM mode (not DPM). This is a CPC-wide setting.

Figure 4-2 expands one or the z/VM partitions from the diagram that is shown in Figure 4-1 on page 84.

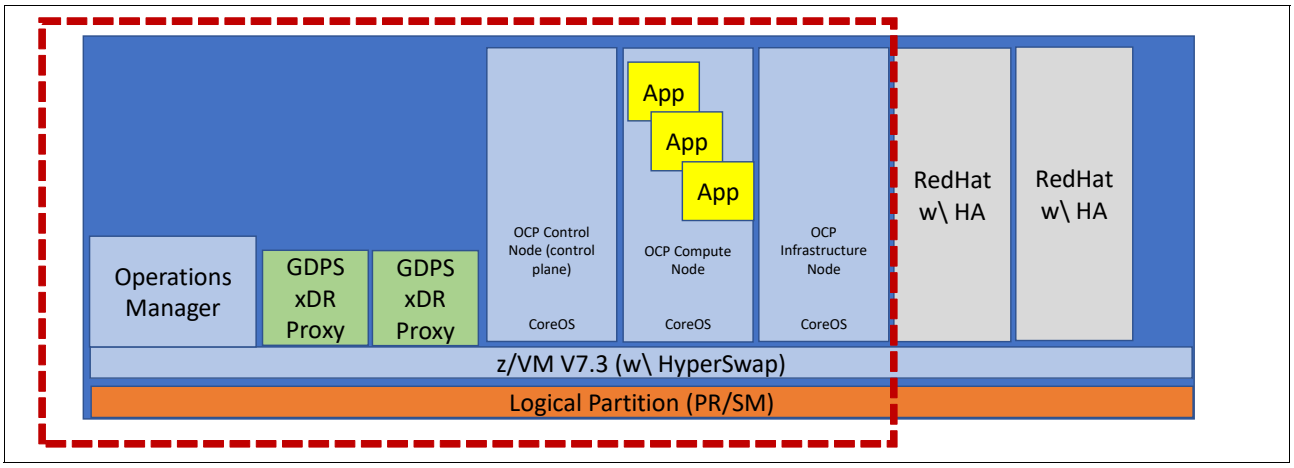


Figure 4-2 Single partition: interior view

Here are some important considerations about the single partition view:

- Red Hat OpenShift Container Platform does not run under Linux on IBM Z. It runs under an IBM Z hypervisor (z/VM or Kernel-based Virtual Machine (KVM)).
- Red Hat Enterprise Linux (RHEL) hosted workloads (or SUSE Linux Enterprise Server) are out of the scope of this example.
- IBM hardware must be in PR/SM mode and not DPM mode.
- z/VM is a requirement to support HyperSwap. The partition contains two xDR proxies.

4.2.2 Network

The network implementation for this scenario is shown in Figure 4-3.

The implementation has multiple (three) virtual devices at the CoreOS level that are connected to a z/VM Layer 2 Virtual Switch. Link aggregation is used to bind multiple Open Systems Adapter (OSA) ports across multiple adapters on a CPC.

An intra z/VM Virtual Switch (VSWITCH) link collates connections across physical hardware in an SSI cluster. Physical switches and cabling are used to bind sites. Sites can be in a 200 km range for Metro Mirror (MM) (GDPS). This implementation enables network traffic separation at virtual, logical, or physical layers that meet security requirements.

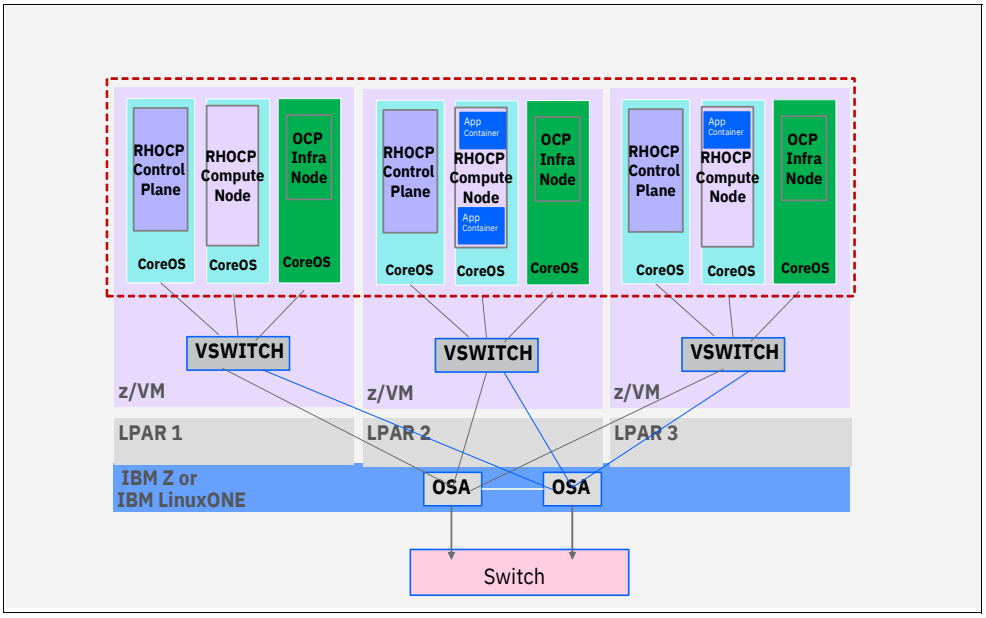


Figure 4-3 Networking diagram

4.2.3 Network optimization

Network optimization is divided in three layers: Red Hat OpenShift Container Platform, z/VM, and physical. Here are the considerations that pertain to each layer:

- Red Hat OpenShift Container Platform considerations:
  - etcd timeout tuning: This timeout determines how long a node goes without hearing a heartbeat before the node attempts to become the leader. The latency for heartbeat control must be within acceptable parameters (a heartbeat interval of 100 ms, and an election timeout of 1000 ms).
  - Heartbeat Interval is the frequency with which the leader notifies followers that it is still the leader. As a best practice, the parameter should be set to the round-trip time between members. By default, etcd uses a 100 ms heartbeat interval.
  - Election Timeout is how long a follower node goes without hearing a heartbeat before the node attempts to become the leader. By default, etcd uses a 1000 ms election timeout.

- z/VM considerations:
  - Analyze site to site network performance looking for VSWITCH congestion.
  - Add network priority to grant control plane traffic first access.
  - Measure the link aggregation to determine optimal port access.
- Physical considerations:
  - GDPS communication is IP-based. An outage there represents a catastrophic network problem, so configure z/OS as cross-site.
  - Network separation:
    - Separate VSWITCHs, or direct-attached OSAs, for sensitive guests.
    - Use VSWITCH Virtual Ethernet Port Aggregator (VEPA) mode to force traffic separation to a physical switch.
  - Optimize local traffic so that critical communications have priority.

4.2.4 Storage optimization

This section describes storage optimization techniques:

- Apps

  Apps are less likely to need persistent or stateful data access.
  - Implementation generally excludes apps, but you should consider which ones you need.
  - You can use Red Hat OpenShift Data Foundation to configure apps for persistent storage instead of network-attached devices, which adds a logical layer with more failover mechanisms (including maintaining multiple replicas of stored data).
  - GDPS can continue to manage ECKD volumes as pertinent, so there is no need for other intervention.
- Red Hat OpenShift Container Platform and z/VM

  Linux-hosted workloads running under z/VM need persistent storage (ECKD on DS8000) that is attached to the guest itself and reflected to the local Linux operating system (OS).
  - Owned by z/VM, and managed by GDPS.
  - You can use HyperSwap on these volumes.
  - You can use z/VM HyperPAV aliasing.
- Red Hat OpenShift Data Foundation

  Red Hat OpenShift Data Foundation is software-defined storage that is optimized for container environments. It runs as an operator on Red Hat OpenShift Container Platform to provide highly integrated and simplified persistent storage management for containers.
  - Red Hat OpenShift Data Foundation enables applications to communicate to persistent storage.
  - Red Hat OpenShift Data Foundation adds replicas of persistent volumes, which are synced and add to the overall resilience of stored data.
  - With ECKD direct-access storage device (DASD), you may restrict the scope of the framework, which eliminates the need to configure network-attached devices.

Figure 4-4 shows the architecture of a single partition configuration with Red Hat OpenShift Data Foundation.

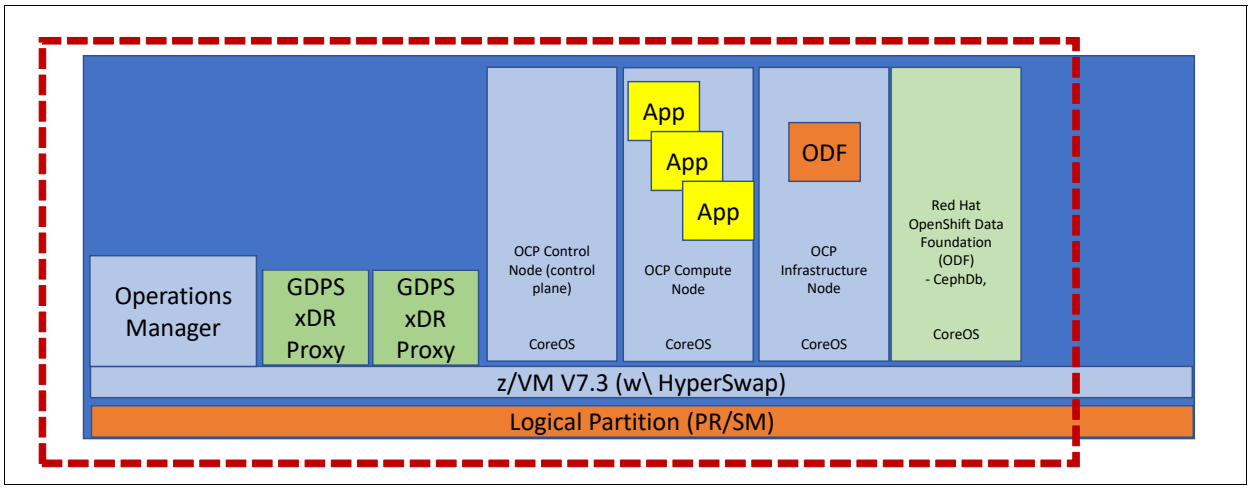


Figure 4-4 Single partition with Red Hat OpenShift Data Foundation

HyperSwap

HyperSwap is an IBM HA solution that provides active-active storage access to volumes that are shared across two physical sites. It is a HA feature that provides dual-site and active-active access to a volume. Data is continuously available at both sites, which improves the availability of your business. HyperSwap is based on synchronous Peer-to-Peer Remote Copy (PPRC) technology for use within a single data center. Data is copied from the primary storage device to a auxiliary storage device.

GDPS Peer-to-Peer Remote Copy (GDPS/PPRC) is an IBM offering that non-disruptively swaps I/O activity from the primary volumes to the secondary volumes of PPRC mirrored pairs when a problem occurs on any primary volume. Automation is used to swap all mirrored volumes concurrently.

Figure 4-5 shows DS8000 storage that is mirrored between Site A and Site B by using PPRC/Metro Mirror (MM). Storage synchronous data replication (mirroring) is the foundational prerequisite for the HyperSwap solution.

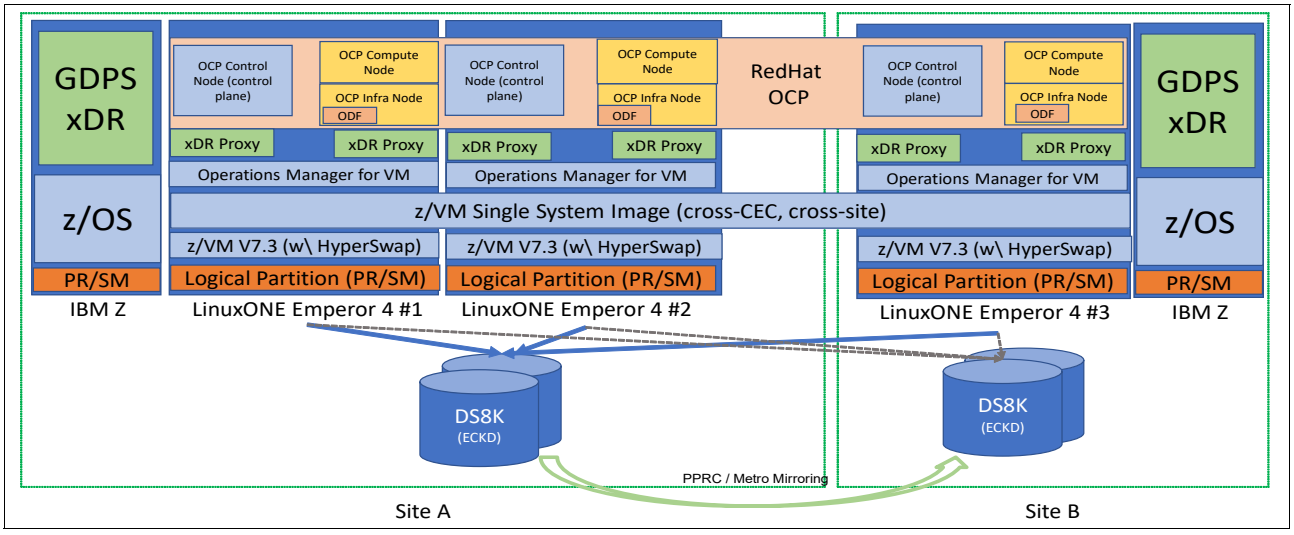


Figure 4-5 PPRC/Metro Mirror, GDPS xDR, and HyperSwap between sites

The blue lines that connect the partitions to the DS8000 disks represent the “in-use” data access paths. The PPRC/MM green arrow represents the required data synchronous replication capability between sites. The gray dotted lines represent the alternative access paths to storage if there is a HyperSwap. HyperSwap can be monitored and controlled by GDPS.

For more information, see “GDPS Virtual Appliance HyperSwap function” on page 54.

4.3 Framework analysis

This section describes the following framework aspects:

- Points of outage
- Potential technical disruption
- Threats
- Mitigations and workaround
- Impacts in a resilient or HA environment and Relative Mean Time to Repair (MTTR)

4.3.1 Points of outage

This section describes points of outage in the framework.

Case 1: Storage unit failure

Table 4-1 describes a case with a storage unit failure.

Table 4-1 Case 1: Storage unit failure

Failing component	What happens	Mitigation, workaround, or recovery
A storage unit fails. It is one DS8000 physical device (not a volume or a link). See Figure 4-6	All data on the storage unit is unavailable.	► Use RAID and data striping to avoid integrity issues. ► GDPS mirrors to a second local storage unit, and PPRC mirrors to a remote storage unit. Use Cyber Vaulting through SafeGuard Copy.
Impact: Negligible. With HyperSwap, storage failover happens within 2 - 3 seconds. (z/VM Switch quiesces now too, so there might be a momentary network blip.)		
Time to recovery of the failing component: 2 hours to restore the physical unit, and 4 hours to uptime. Moore time to mirror current data to the restored volume is required. (The business impact is not included in the calculation.)		

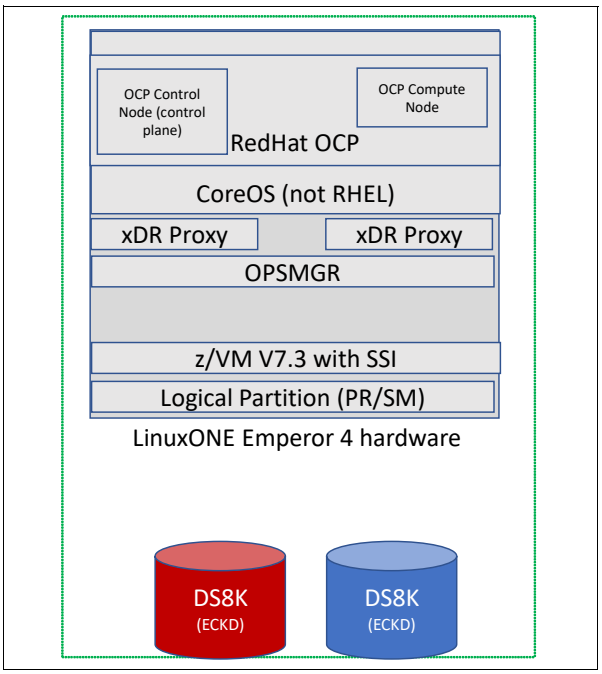


Figure 4-6 Storage unit failure

Case 2: Hardware unit outage

Table 4-2 describes a case with a hardware unit outage.

Table 4-2 Case 2: Hardware unit outage

Failing component	What happens	Mitigation, workaround, or recovery
Hardware unit outage (one LinuxONE IV CPC in entirety). See Figure 4-7.	Compute for an entire system (1-n logical partitions (LPARs) and hosted workload) is not available.	► Use core sparing for general workload bypass. ► Use RAIM to avoid memory corruption. ► Use physical redundancy inside a CPC to mitigate problems.
Impact: Negligible. The workload continues on other CPCs autonomically.		
Time to recovery of the failing component: 4 hours is typical for CPC repair. There is no data rebuild in this case (it is compute).		

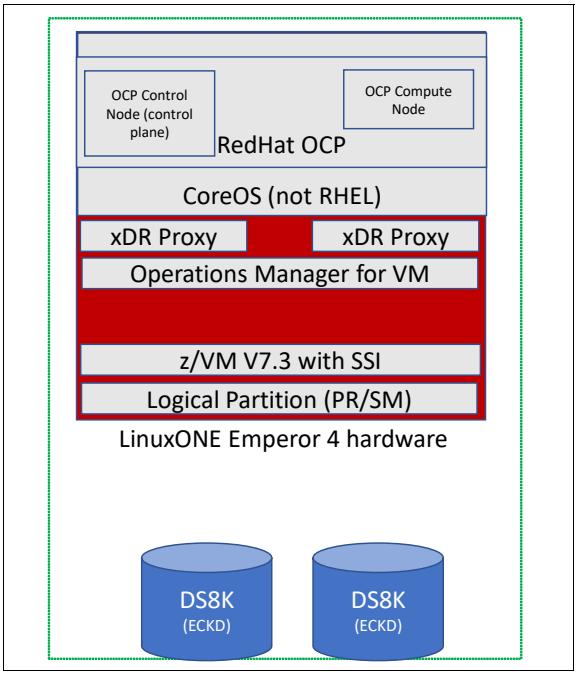


Figure 4-7 Hardware unit failure

Case 3: Logical partition goes into a “wait state” or has an outage

Table 4-3 describes a case where an LPAR goes into a “wait state” or has an outage.

Table 4-3 Case 3: Logical partition goes into a “wait state” or has an outage

Failing component	What happens	Mitigation, workaround, or recovery
LPAR goes into a wait state or has an outage. See Figure 4-8.	The compute for one partition and its hosted workload is gone.	► Use SIE isolation to prevent impacts to other workloads. ► Reschedule the workload on to other compute nodes (Red Hat OpenShift).
Impact: Negligible. Similar to hardware, z/VM and Red Hat OpenShift Container Platform workloads may move to other partitions in the SSI or systems in the cluster.		
Time to recovery of the failing component: N/A. LPAR instantiation is near-immediate, and failure at the LPAR level almost always means “workload”.		

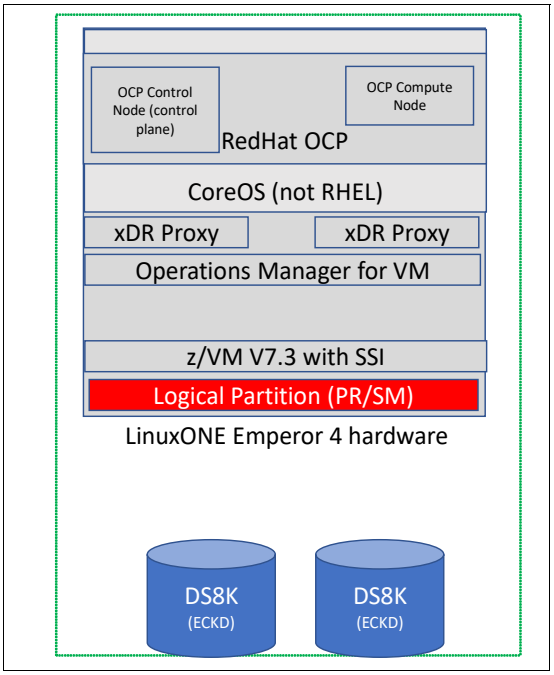


Figure 4-8 Logical partition outage

Case 4: One z/VM system has an outage

Table 4-4 describes a case where one z/VM system has an outage.

Table 4-4 Case 4: One z/VM system has an outage

Failing component	What happens	Mitigation, workaround, or recovery
One z/VM system has an outage. See Figure 4-9.	The virtual infrastructure is gone (for one LPAR).	► Use Operations Manager to automate general use-case recovery. ► Use SSI and relocation (if it is a planned outage). ► Perform Red Hat OpenShift Container Platform rescheduling.
Impact: Negligible. SSI enables guest mobility to relocate workloads. Red Hat OpenShift Container Platform reschedules work onto other compute resources.		
Time to recovery of the failing component: 5 minutes to reactivate the partition and start the z/VM system. A longer recovery time past varies based on the number of guests starting, and the size of the z/VM partition (which impacted the time to take the dump that is used in problem determination). A start of the z/VM system includes guests, virtual storage, GDPS xDR proxies, virtual networking, and Red Hat OpenShift Container Platform for that system. It does not include resync operations of various layers.		

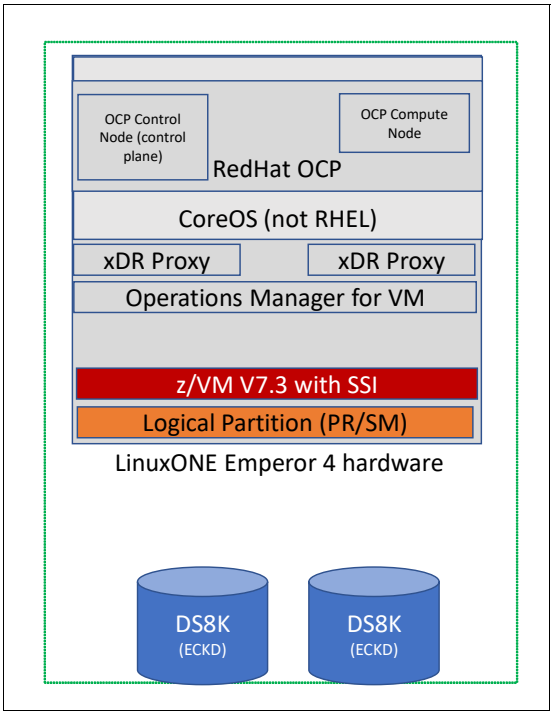


Figure 4-9 One z/VM system has an outage

Case 5: A z/VM virtual network encounters a problem

Table 4-5 describes a case where a z/VM virtual network encounters a problem.

Table 4-5 Case 5: A z/VM virtual network encounters a problem

Failing component	What happens	Mitigation, workaround, or recovery
A z/VM virtual network encounters a problem. See Figure 4-10.	Traffic for z/VM or guests is disrupted, which leads to problems with heartbeats and system assessment. There is disruption to the client workload.	► Use link aggregation to collect multiple OSA ports into a common logical channel. ► Set up VSWITCH automatic failover (four controller nodes) in case of a disrupted network operation. ► Use an Inter-VSWITCH Link to coordinate multiple z/VM systems.
Impact: Negligible. Link aggregation collates resources, and failover prevents an outage.		
Time to recovery of the failing component: If it is physical networking, the time to recover varies. If it is virtual networking, expect 1 - 2 hours to debug and rebuild.		

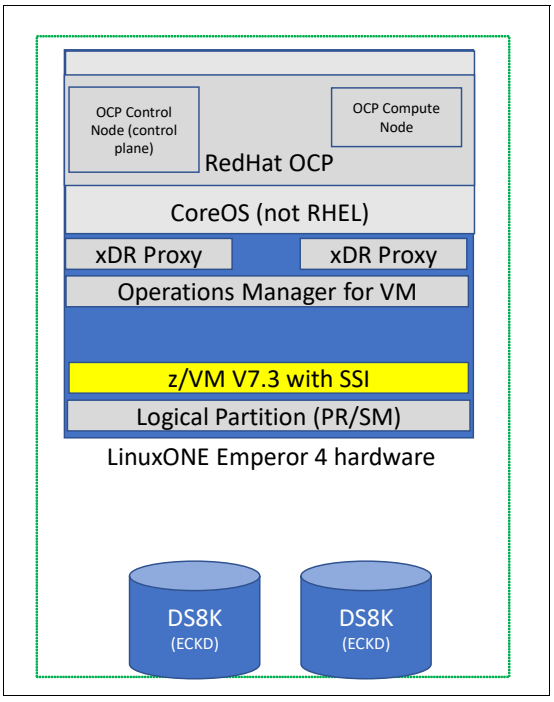


Figure 4-10 z/VM network, I/O, or SSI problem

Case 6: z/VM I/O links or channels take an outage

Table 4-6 describes a case where there is an outage of z/VM I/O links or channels.

Table 4-6 Case 6: z/VM I/O links or channels take an outage

Failing component	What happens	Mitigation, workaround, or recovery
z/VM I/O links or channels take an outage. See Figure 4-10 on page 94.	Access to storage through the virtual infrastructure is disrupted. Data cannot be read in or written out, which can potentially disrupting business.	► Use multiple channel paths for access to storage units. ► Use Operations Manager for local scripting to mitigate failures.
Impact: Negligible. Data remains available even with a failing virtual or logical device. (If the last channel path standing fails, a HyperSwap is triggered.)		
Time to recovery of the failing component: Varies. Depends on whether the failure is physical or logical. (2 - 3 seconds if it is a restart of a channel path. It can take hours if there is a physical problem or cabling issue.)		

Case 7: One z/VM system has an outage

Table 4-7 describes a case where one z/VM system has an outage.

Table 4-7 Case 7: One z/VM system has an outage

Failing component	What happens	Mitigation, workaround, or recovery
A z/VM SSI has a one-node failure. See Figure 4-10 on page 94.	Individual systems keep running, but guest mobility is no longer feasible. There is potential for split-brain problems in zoned workloads that are running under z/VM (Red Hat OpenShift Container Platform).	► Put systems in SAFE mode to continue operating (manual intervention). ► SSI members do not shoot the other node in the head (STONITH) ► Red Hat OpenShift Container Platform should STONITH. ► SSI rejoins missing members after repairs are made and the state is reaffirmed.
Impact: Potential disruptions to Red Hat OpenShift Container Platform quorum and workload dispatch if SAFE-mode system remains unavailable. However, the workload is soon rebalanced.		
Time to recovery of the failing component: For more information, see “Case 4: One z/VM system has an outage” on page 93 The recovery of one system varies depending on the damage. The time to start remains constant.		

Case 8: z/VM SSI takes a hit to its Persistent Data Record

Table 4-8 describes a case where a z/VM SSI takes a hit to its Persistent Data Record.

Table 4-8 Case 8: z/VM SSI takes a hit to its Persistent Data Record

Failing component	What happens	Mitigation, workaround, or recovery
A z/VM SSI takes a hit to its Persistent Data Record. See Figure 4-11	The Persistent Data Record volume maintains the state of the overall cluster and heartbeat tracking. If the cluster is damaged, the whole cluster moves into recovery mode.	Regarding GDPS, the PDR is not handled separately from the rest of the GDPS consistency group.
Impact: Minimal. HyperSwap takes 2 - 3 seconds. Persistent Data Record becomes a single point of failure (SPOF) until the original Persistent Data Record volume is restored.		
Time to recovery of the failing component: After the storage device is repaired (see “Case 1: Storage unit failure” on page 90), MM resumes active backup.		

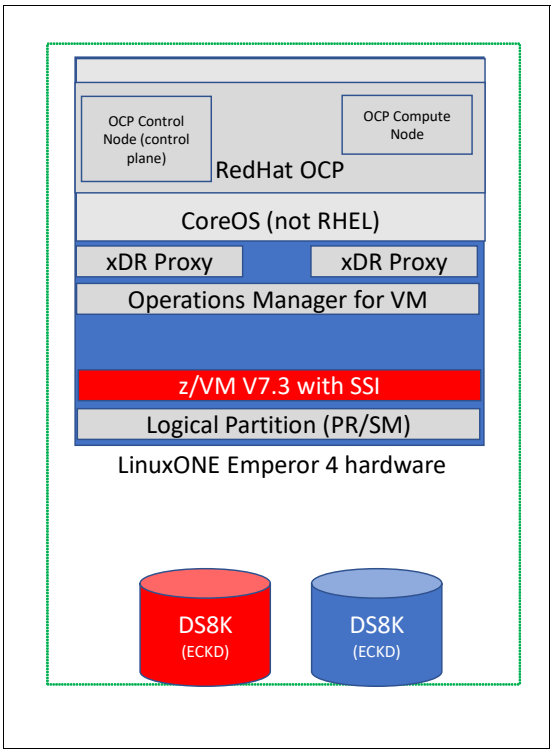


Figure 4-11 SSI Persistent Data Record takes a hit

Case 9: xDR Proxy failure

Table 4-9 describes a case where there is a xDR Proxy failure.

Table 4-9 Case 9: xDR Proxy failure

Failing component	What happens	Mitigation, workaround, or recovery
The xDR Proxy fails. See Figure 4-12.	The xDR satellite fails on a z/VM system, which inhibits data replication across sites.	There is an automatic failover to back up the xDR satellite on the z/VM system.
Impact: None. The xDR Proxy failover is immediate.		
Time to recovery of the failing component: 2 - 3 seconds. 2 - 3 minutes to restart the xDR proxy guest.		

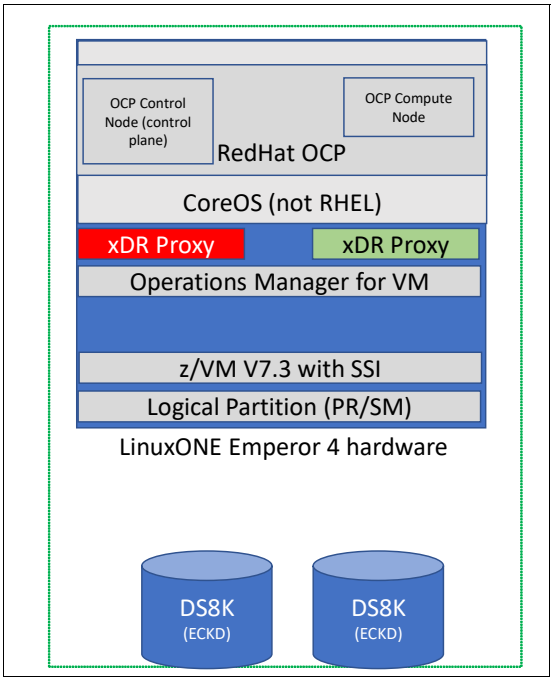


Figure 4-12 xDR Proxy failure

Case 10: Network disruption: no xDR Proxy access to IBM Z GDPS

Table 4-10 describes a case where there is a network disruption because there is no xDR Proxy access to IBM Z GDPS.

Table 4-10 Case 10: Network disruption: no xDR Proxy access to IBM Z GDPS

Failing component	What happens	Mitigation, workaround, or recovery
A networking disruption prevents xDR Proxy access to an IBM Z GDPS installation. See Figure 4-13	Commands from z/OS are inhibited in terms of making a backup.	► If you are running z/OS in a nearby partition, this disruption is not likely to be an issue (HiperSockets and local switches). ► For more information, see IBM Z with z/OS Resilience Best Practices Guide: Experienced Technical Architect or Designer.
Impact: N/A. Because GDPS is IP-based, this situation happens only if an entire network is destroyed.		
Time to recovery of the failing component: N/A.		

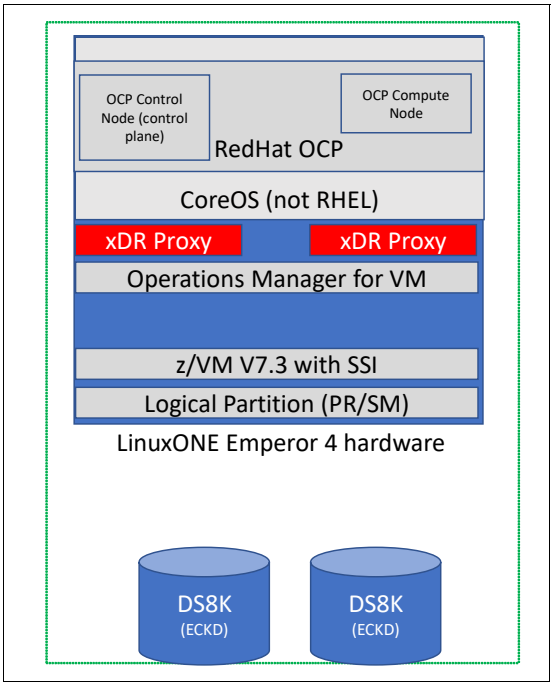


Figure 4-13 xDR connectivity issue

Case 11: CoreOS system fails

Table 4-11 describes a case where the CoreOS system fails.

Table 4-11 Case 11: CoreOS system fails

Failing component	What happens	Mitigation, workaround, or recovery
The CoreOS system fails. See Figure 4-14.	The z/VM system is running, but a Red Hat guest has a kernel panic, which disrupts the workload.	► Red Hat OpenShift Container Platform checks the control status and reschedules the workload if needed. ► z/VM restarts if needed (automatable).
Impact: Temporary loss of compute that is related to control or compute. The workload is redistributed or restarted on other nodes as needed.		
Time to recovery of the failing component: N/A. There is not a CoreOS instantiation independent of Red Hat OpenShift Container Platform operations, so this event is not a distinct point of outage from a Red Hat OpenShift Container Platform crash.		

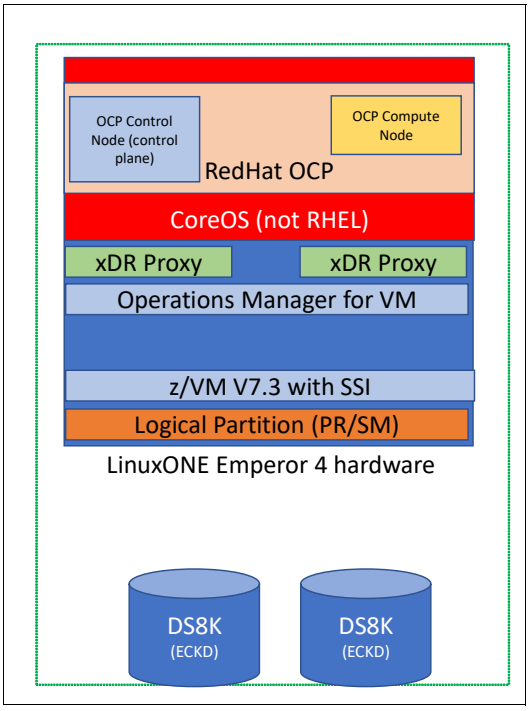


Figure 4-14 CoreOS System crash

Case 12: CoreOS networking failure inside a Linux guest

Table 4-12 describes a case where there is a CoreOS networking failure inside a Linux guest.

Table 4-12 Case 12: CoreOS networking failure inside a Linux guest

Failing component	What happens	Mitigation, workaround, or recovery
There is a CoreOS networking failure inside a Linux guest. See Figure 4-15.	The Linux instance has a networking device failure.	The workload is rescheduled or restarted on other operational nodes. After the network recovery node is accessible, workloads can be scheduled.
Impact: Temporary loss of compute that is related to control or compute. The workload is redistributed or restarted on other nodes as needed.		
Time to recovery of the failing component: N/A. There is not a CoreOS instantiation that is independent of Red Hat OpenShift Container Platform operations, so this event is not a distinct point of outage from a Red Hat OpenShift Container Platform failure.		

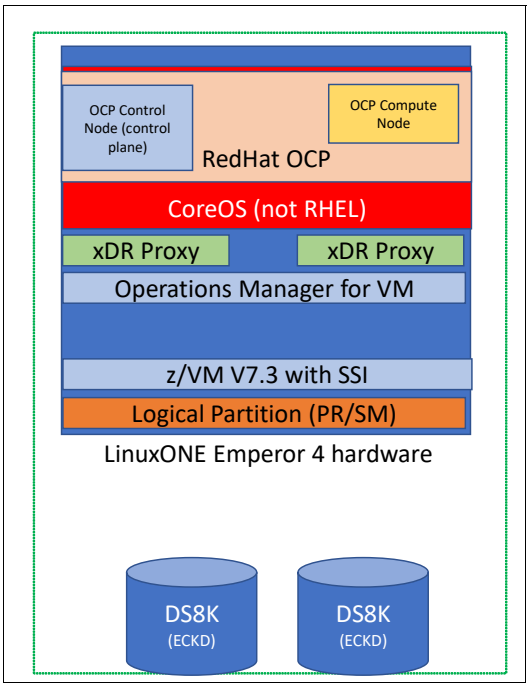


Figure 4-15 CoreOS networking failure inside a Linux guest

Case 13: CoreOS loses storage devices

Table 4-13 describes a case where CoreOS loses storage devices.

Table 4-13 Case 13: CoreOS loses storage devices

Failing component	What happens	Mitigation, workaround, or recovery
CoreOS loses storage devices. See Figure 4-15 on page 100.	A Linux instance loses access to necessary storage for hosted applications.	► Linux continues to run and can save data to the page cache. When the device is recovered, the data is stored on disk. ► If a guest fails, restart the guest after a storage connection is reestablished. ► If storage is invalid, reinstall the node.
Impact: Temporary loss of compute related to control or compute. The workload is redistributed or restarted on other nodes as needed.		
Time to recovery of the failing component: N/A. There is not really a CoreOS instantiation independent of Red Hat OpenShift Container Platform operations, so this event is not a distinct point of outage from a Red Hat OpenShift Container Platform failure.		

Case 14: Red Hat OpenShift Container Platform control-node failure

Table 4-14 describes a case where there is a Red Hat OpenShift Container Platform control-node failure.

Table 4-14 Case 14: Red Hat OpenShift Container Platform control-node failure

Failing component	What happens	Mitigation, workaround, or recovery
There is a Red Hat OpenShift Container Platform Control-node failure. See Figure 4-16.	A Linux instance loses access to necessary storage for hosted applications.	► The workload continues to operate, but the cluster shows a failure state. ► The workload is rescheduled on other operational nodes.
Impact: Temporary loss of compute related to control for certain programs. The workload is redistributed or restarted on other nodes as needed.		
Time to recovery of the failing component: 30 - 60 minutes if there are automation and backups. (The system administrator must reinstall Red Hat OpenShift Container Platform from scratch and have the new instantiation of the node rejoin the cluster. There is no data recovery by itself.)		

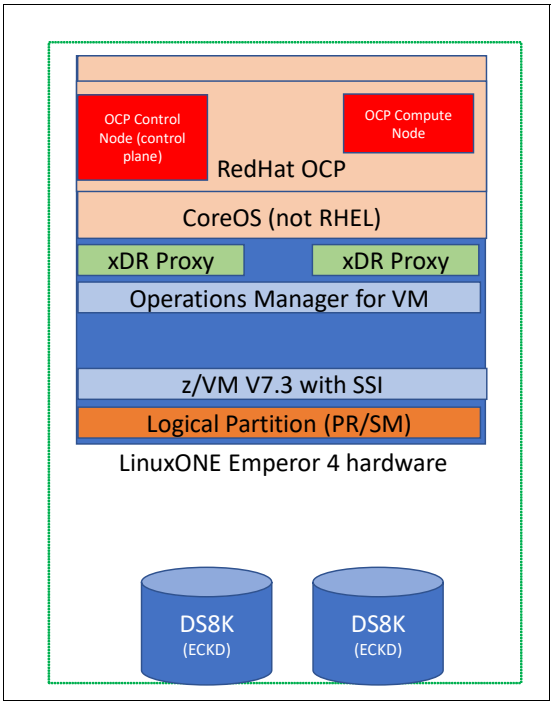


Figure 4-16 Red Hat OpenShift Container Platform control-node failure

Case 15: Red Hat OpenShift Container Platform Compute node restart control-node failure

Table 4-15 Case 15: Red Hat OpenShift Container Platform Compute node restart control-node failure

Failing component	What happens	Mitigation, workaround, or recovery
Red Hat OpenShift Container Platform Compute node does not restart. Figure 4-16 on page 102.	See Impact below.	The workload is rescheduled or restarted on other operational nodes. After the network recovery node is accessible, workloads can be scheduled.
Impact: Temporary loss of compute related to control for certain programs. The workload is redistributed or restarted on other nodes as needed.		
Time to recovery of the failing component: 30 - 60 minutes if there are automation and backups. (The system administrator must reinstall Red Hat OpenShift Container Platform from scratch and have the new instantiation of the node rejoin the cluster. There is no data recovery by itself.)		

4.3.2 Planned outages

This section describes the application of service or new-release installation procedure as a potential disruption to availability at each level of the LinuxONE availability model.

Case 1: Planned outage for a CPC or a storage unit

Table 4-16 describes a case where there is a planned outage for a CPC or a storage unit.

Table 4-16 Case 1: Planned outage for CPC or storage unit

Failing component	What happens	Mitigation, workaround, or recovery
There is a planned outage for a CPC or a Storage Unit. See Figure 4-17.	Hardware is down or the firmware is down for the Microcode (or Machine) Change Level application.	► z/VM relocates work to alternative hardware, and Red Hat OpenShift Container Platform redistributes compute to available nodes. There is no outage. ► IBM HyperSwap sends the workload to an alternative storage volume. There is no outage.
There is a recovery error on a CPC or a storage unit.	The hardware remains down.	The workload persists on alternative systems until the CPC or storage unit is repaired. There is no further outage because the workload never rebalanced.

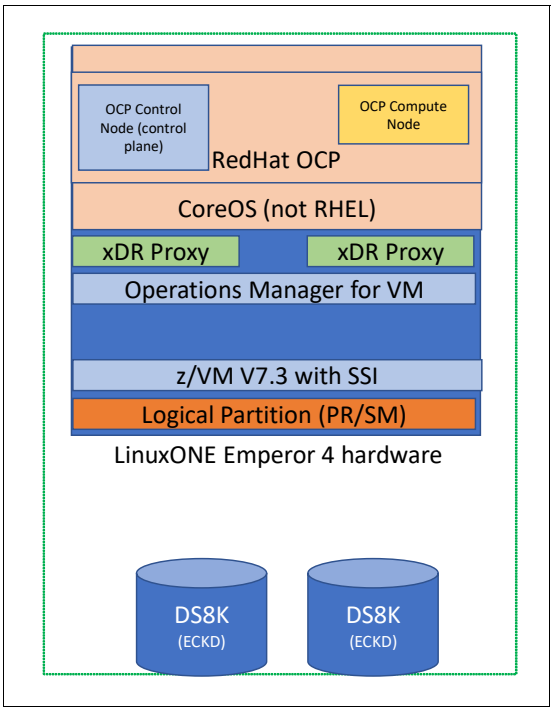


Figure 4-17 Planned outage for CPC or storage

Case 2: Planned outage for z/VM

Table 4-17 describes a case where there is a planned outage for z/VM.

Table 4-17 Case 2: Planned outage for z/VM

Failing component	What happens	Mitigation, workaround, or recovery
There is a planned outage for z/VM. See Figure 4-17.	One z/VM system is down for the PTF application.	The z/VM admin (or GDPS) relocates work to alternative z/VM systems. Red Hat OpenShift Container Platform redistributes compute to available nodes. There is no outage.
Time to recovery of the failing component: ► 4 hours to complete a new installation of z/VM throughout an SSI. ► 5 minutes to install a service PTF on one member of an SSI. ► Assumes that all planning is done in advance.		
Recovery error on a CPC or a storage unit.	z/VM cluster goes into SAFE mode and isolates one system.	There is no outage, but there are communication problems with the workload that is running in the SAFE system until a repair finishes.
For more information, see the following resources: ► Removing service ► z/VM Education: How to remove service		

Case 3: Planned outage for Red Hat OpenShift Container Platform

Table 4-18 describes a case where there is a planned outage for Red Hat OpenShift Container Platform.

Table 4-18 Case 3: Planned outage for Red Hat OpenShift Container Platform

Failing component	What happens	Mitigation, workaround, or recovery
There is a planned outage for Red Hat OpenShift Container Platform. See Figure 4-17 on page 104.	There was a restart of a control plane. (The service was applied in the background.)	► There is no outage. ► GDPS can initiate guest mobility and z/VM servers can be relocated (VMRELOCATE).
There is a recovery error on Red Hat OpenShift Container Platform.	Problems with cluster (control, compute)	► On control nodes, there is no outage up to a single node down. If two control nodes are down, the workloads still run, but in a degraded mode until the failed nodes recover. ► On compute nodes, there is no outage if other compute nodes are available and the application is scaled up on all running compute nodes. For more information, see 5.1, “About disaster recovery”.

Case 4: Planned outage for GDPS xDR

Table 4-19 describes a case where there is a planned outage for GDPS xDR.

Table 4-19 Case 4: Planned outage for GDPS xDR

Failing component	What happens	Mitigation, workaround, or recovery
There is a planned outage for Red Hat OpenShift Container Platform. See Figure 4-17 on page 104.	N/A.	► There is no outage. ► xDR proxies can be recycled one at a time after maintenance. The system is never without a local agent. ► Similarly, multiple z/OS K-systems update one at a time, but never without GDPS xDR.
There is a recovery error on GDPS xDR.	There are problems with storage synchronization, and there is a lack of failover support until the problem is mitigated.	► Recycle or restart. ► No known cases of this situation occurring.

4.4 Disaster recovery

This section describes disaster scenarios, such as the loss of entire data center or regions due to natural disasters, power outages, or human-generated catastrophes; and other events that can cause availability loss.

Generally, physical data centers (and their full recovery) are measured at five 9's of availability (99.999%).

- Data center providers have service-level agreements (SLAs) regarding availability and resources.
- Calculating availability, which is based on data centers worldwide, is less reliable than for hardware or software.
- Standards vary in adoption and locality, for example, ANSO/BICSI 002-2019.
- Use a tiered approach based on how much a client can tolerate (or afford).

For less extreme problems (water, electricity, or air conditioning outages), data centers should provide duplication of resource, and have backup generators, alternative lines, and disparate power grids (for maintenance and planned outages). The rate of failure should be low, but depending on the nature of the disaster, restoring the physical assets might be difficult or impossible.

Having geographically dispersed data centers minimizes the chance of a single disaster hitting both data centers simultaneously. Having data replication through GDPS is important whether the presented framework is configured Active-Active.

Figure 4-18 shows an example of an entire site data center (Site B) outage.

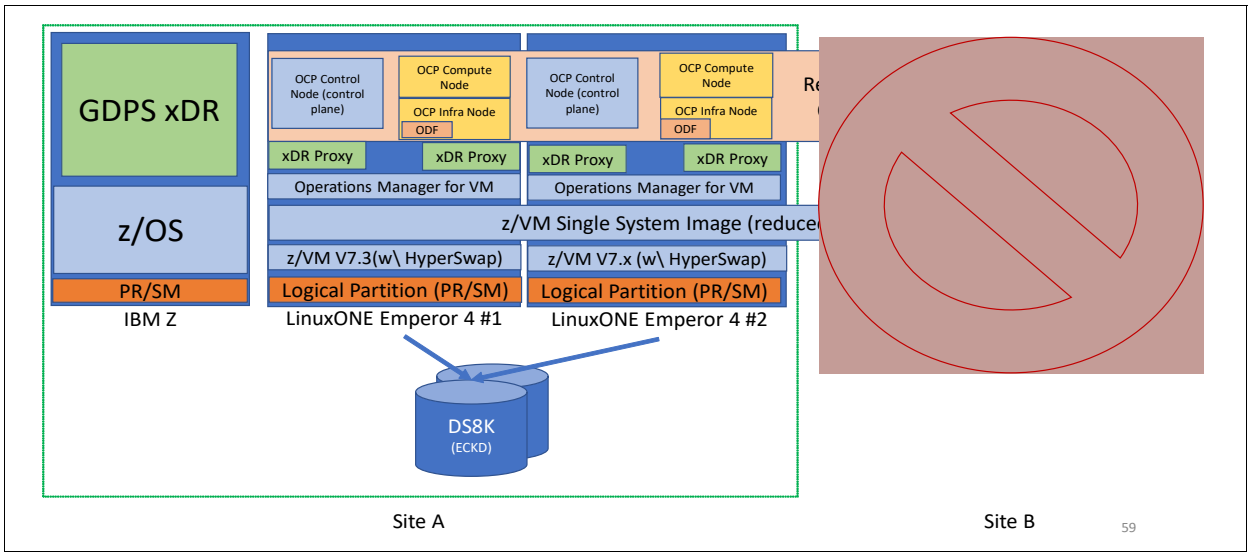


Figure 4-18 Site B has a data center outage

Case 1: Data center outage

Table 4-20 describes a case where there is a data center outage.

Table 4-20 Case 1: Data center outage

Failing component	What happens	Mitigation, workaround, or recovery
An entire data center had an outage. See Figure 4-18 and Figure 4-19.	There is a loss of all compute, network, and storage in that building.	► Have redundant electrical, water, power, and infrastructure to keep the data center running. ► GDPS restarts the virtual machines (VMs) on the surviving cluster members, or restarts z/VM in new partitions. ► GDPS sends compute to the surviving storage. ► z/VM can restart VMs on the surviving cluster members. ► Red Hat OpenShift Container Platform redirects the workload to surviving compute nodes when it detects a loss of compute or control VMs.
Impact: Reduced “white space” overhead for future failover, and reduced redundancy opportunities for compute and storage. There is a change in the network latency patterns. ► Because it is an unplanned outage, restarting multiple systems might take time. ► Red Hat OpenShift Container Platform restarts applications (or fail over to backups) as needed. ► z/VM Operations Manager must trigger VM IPL (minutes). **Time to recovery of the failing component:** Depends on the nature of the disaster. It might be days, weeks, or years. The workload can continue, but at a greater risk if other problems occur.		

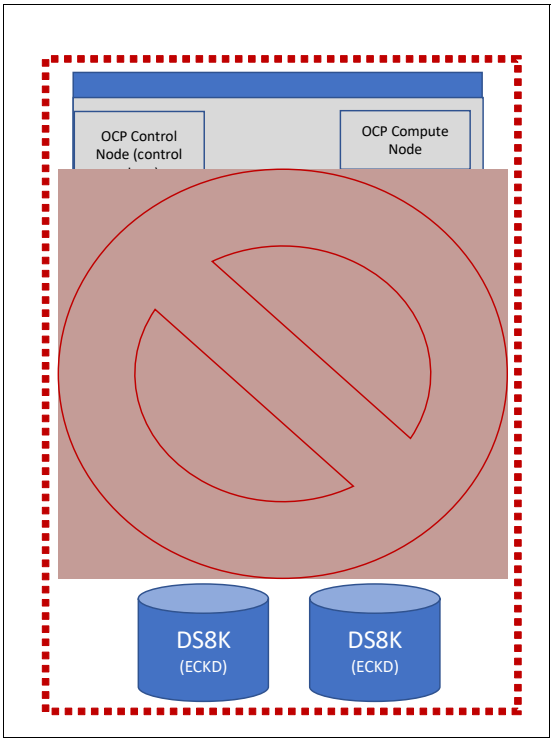


Figure 4-19 Data center outage

The framework that was covered in this chapter illustrates continuous operation of a Red Hat OpenShift Container Platform workload in the context of about three partitions that are backed by GDPS xDR:

- A robust DR plan often has reserve LPARs that are available if the framework does not consume three full CPCs.
- Other partitions are under use for other work, such as development work or test work.
- With the correct system configuration, a z/VM SSI member can start in a new location.

Based on availability and disaster recovery (DR) requirements, consider having a failover plan for z/VM SSI member nodes:

- Mitigates the damage that another outage might incur.
- Lessens the time that an SSI may run closer to a memory “ceiling”.
- Partition failover enables Red Hat OpenShift Container Platform to run at “full strength” while a physical site is under repair.

4.5 Resources

For more information about the topics that were described in this chapter, see the following resources:

- *Building a Red Hat OpenShift Environment on IBM Z*, SG24-8515
- *Getting Started with IBM Z Resiliency*, SG24-8446
- *End-to-End High Availability Solution for System z from a Linux Perspective*, SG24-8233
- *IBM GDPS: An Introduction to Concepts and Capabilities*, SG24-6374
- SUSE Linux Enterprise Server 11 SP2 High Availability Guide
- What is IT service management (ITSM)?
- Solution papers

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.

IBM Redbooks

The following IBM Redbooks publications provide more information about the topics in this document. Some publications that are referenced in this list might be available in softcopy only.

- *Getting Started with IBM Z Resiliency*, SG24-8446
- *IBM GDPS: An Introduction to Concepts and Capabilities*, SG24-6374
- *IBM z16 Configuration Setup*, SG24-8960
- *IBM Z Connectivity Handbook*, SG24-5444

You can search for, view, download, or order these documents and other Redbooks, Redpapers, web docs, drafts, and additional materials, at the following website:

ibm.com/redbooks

Online resources

- *Hardware Management Console (HMC): Version 2.16.0 - 11 July 2024*:

  https://www.ibm.com/docs/en/module_1675371155154/pdf/HMCVersion2160-11July2024.pdf

- IBM Resource Link:

  http://www.ibm.com/servers/resourcelink

Help from IBM

IBM Support and downloads

ibm.com/support

© Copyright IBM Corp. 2024. All rights reserved.

Abbreviations and acronyms

AIS	Application Interface Specification
BC	business continuity
CA	continuous availability
CADR	continuous availability and disaster recovery
CBU	Capacity BackUp
CHPID	channel path identifier
CIB	Cluster Information Base
CIU	Customer Initiated Upgrade
CKD	count-key data
CO	continuous operations
CoD	Capacity on Demand
CORBA	Common Object Request Broker Architecture
CP	central processor or Control Program
CPC	central processor complex
CRMd	Cluster Resource Management daemon
CRO	continuous reliable operation
CSM	Copy Services Manager
CTC	channel-to-channel
DASD	direct-access storage device
DC	Designated Coordinator
DIMM	Dual Inline Memory Module
DIX	Data Integrity Extensions
DORA	Digital Operational Resilience Act
DPM	Dynamic Partition Manager
DR	disaster recovery
DRAM	dynamic random access memory
EAL	Evaluation Assurance Level
EAL5	Evaluation Assurance Level 5
EAV	Extended Address Volume
ECKD	Extended Count Key Data
EDA	Enhanced Drawer Availability
FBA	Fixed-Block Architecture
FC	Fibre Channel
FCP	Fibre Channel Protocol
FICON	Fibre Channel connection
GM	Global Mirror
HA	high availability
HADR	high availability and disaster recovery
HCD	Hardware Configuration Definition
HMA	Hardware Management Appliance
HMC	Hardware Management Console
HPI	Hardware Platform Interface
IDAA	IBM Db2 Analytics Accelerator
IFL	Integrated Facility for Linux
IFP	Integrated Firmware Processor
IOCDS	Input/Output Configuration Data Set
IOCP	Input/Output Configuration Program
ISFC	Inter-System Facility for Communications
ISV	independent software vendor
KVM	Kernel-based Virtual Machine
LACP	Link Aggregation Control Protocol
LAN	local area network
LCSS	logical channel subsystem
LCU	Logical Control Unit
LGR	Live Guest Relocation
LICCC	Licensed Internal Code Configuration Control
LPAR	logical partition
LPN	LinuxONE Partner Network
LSS	logical subsystem
LUN	logical unit number
MGM	Metro Global Mirror
MM	Metro Mirror
MOM	message-oriented middleware
MTTR	Mean Time to Repair
NIC	network interface card
NPIV	N_Port ID Virtualization
OCI	Open Container Initiative
OOCoD	On/Off Capacity on Demand
ORB	object request broker
OS	operating system
OSA	Open Systems Adapter
OSA/SF	Open System Adapter Support Facility
OSC	oscillator card

© Copyright IBM Corp. 2024.

PAV	Parallel Access Volume
PHS	Planned HyperSwap
PPRC	Peer-to-Peer Remote Copy
PU	processing unit
QDIO	queued direct input/output
RAIM	Redundant Array of Independent Memory
RAS	reliability, availability, and serviceability
RDMA	Remote Direct Memory Access
REST	Representational State Transfer
RHEL	Red Hat Enterprise Linux
RoCE	Remote Direct Memory Access over Converged Ethernet
RPC	remote procedure call
RPO	recovery point objective
RSF	Remote Support Facility
RTO	recovery time objective
SA MP	System Automation for Multiplatforms
SAF	Service Availability Forum
SAN	storage area network
SAP	system assist processor
SE	Support Element
SFS	Shared File System
SLA	service-level agreement
SMC	Shared Memory Communication
SMC-D	Shared Memory Communications-Direct Memory Access
SMC-R	Shared Memory Communications over Remote Direct Memory Access
SNA	System Network Architecture
SPOF	single point of failure
SS	subchannel set
SSC	Secure Service Container
SSCH	Start Subchannel
SSI	Single System Image
STONITH	Shoot the Other Node in the Head
SVC	SAN Volume Controller
TPM	transaction processing monitor
TSL	Transport Layer Security
UHS	Unplanned HyperSwap
VEPA	Virtual Ethernet Port Aggregator
VM	virtual machine
VSWITCH	z/VM Virtual Switch
xDR	Extended Disaster Recovery

Redbooks

IBM LinuxONE Resiliency

(0.2”spine)
0.17”<->0.473”
90<->249 pages



SG24-8544-00

ISBN 0738461679

Printed in U.S.A.

Get connected



