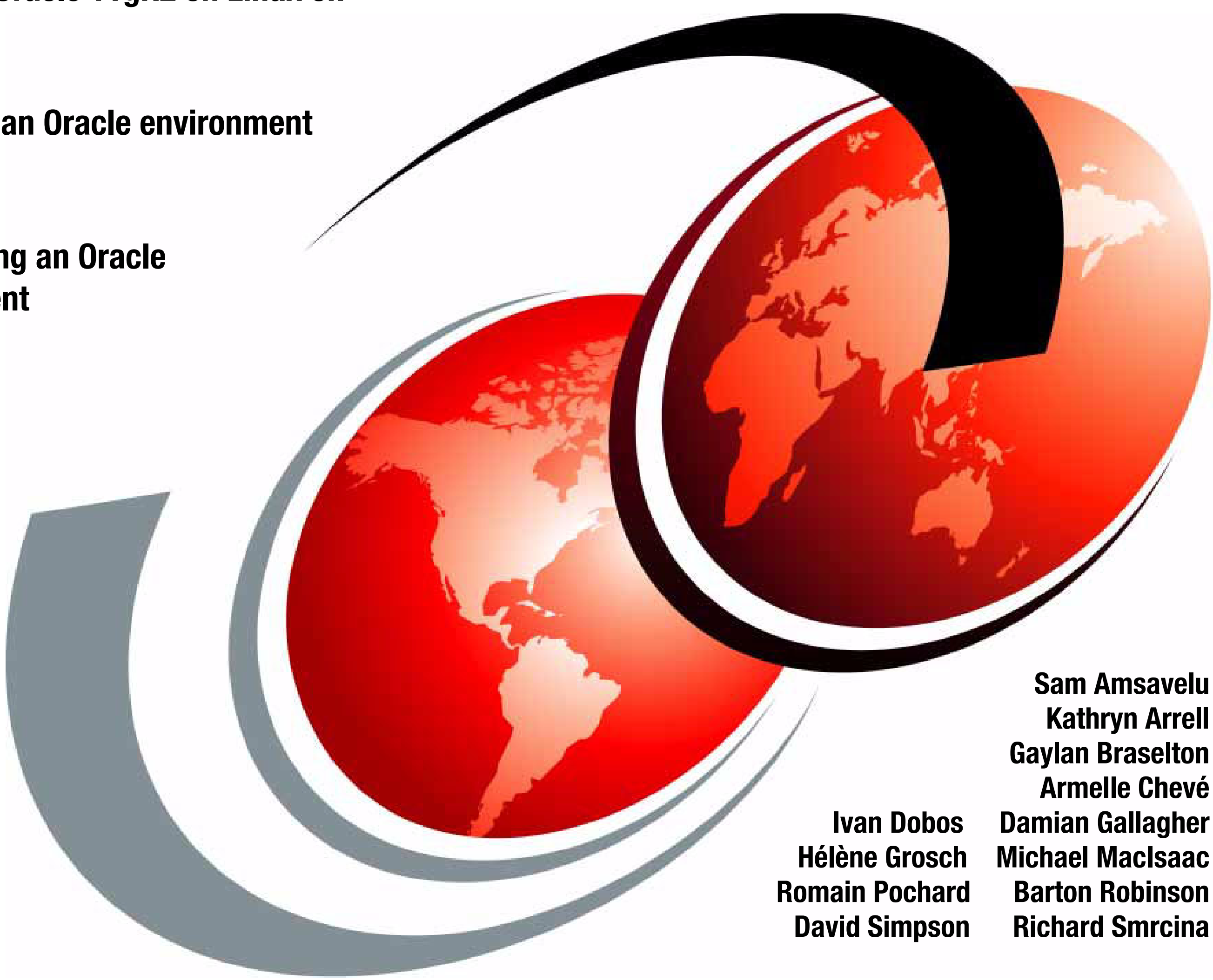
IBM

Experiences with Oracle 11gR2 on Linux on System z

Installing Oracle 11gR2 on Linux on System z

Managing an Oracle environment

Provisioning an Oracle environment



Sam Amsavelu
Kathryn Arrell
Gaylan Braselton
Armelle Chevé
Ivan Dobos
Damian Gallagher
Hélène Grosch
Michael MacIsaac
Romain Pochard
Barton Robinson
David Simpson
Richard Smrcina

Redbooks

ibm.com/redbooks

International Technical Support Organization

Experiences with Oracle 11gR2 on Linux on System z

November 2013

SG24-8104-00

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

First Edition (November 2013)

This edition applies to Version 11gR2 of the Oracle Database.

Contents

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

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

AIX®
BladeCenter®
CloudBurst®
DB2®
DirMaint™
ECKD™
FICON®
Global Business Services®
HiperSockets™
IA®
IBM®
IMS™
InfoSphere®
POWER7®
RACF®
Redbooks®
Redpaper™
Redpapers™
Redbooks (logo) ®
S/390®
Smarter Planet®
System p®
System x®
System z®
Tivoli®
Velocity™
WebSphere®
XIV®
z/OS®
z/VM®
z/VSE®
z10™
zEnterprise®
zSeries®

The following terms are trademarks of other companies:

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

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

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

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

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

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

Preface

Linux on System z offers many advantages to customers who rely on the IBM® mainframe systems to run their businesses. Linux on System z makes use of the qualities of service in the System z® hardware and in z/VM®, making it a robust industrial strength Linux. This provides an excellent platform for hosting Oracle solutions that run in your enterprise.

This IBM Redbooks® publication is divided into several sections to share the following experiences that are gained while Oracle Database 11gR2 is installed and tested:

- Setting up Red Hat Enterprise Linux 6 for Oracle
- Managing an Oracle on Linux on System z environment
- Provisioning Linux guests using several tools

It also includes many general hints and tips for running Oracle products on IBM System z with Linux and z/VM.

Interested readers include database consultants, installers, administrators, and system programmers. This book is not meant to replace Oracle documentation but to supplement it with our experiences while Oracle products are installed and used.

Authors

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

Sam Amsavelu is a Certified Consulting IT Architect in the IBM Advanced Technical Support Organization, supporting Oracle on System z Linux and Siebel on System z customers. He has more than 25 years of IT experience in IBM products.

Kathryn Arrell is an Oracle Specialist at the IBM/Oracle International Competency Center at IBM San Mateo. Previously, she worked as an ERP specialist at the ITSO in Poughkeepsie, New York.

Gaylan Braselton is Sales Specialist for System z Oracle Solutions and responsible for System z Sales with Oracle for North America. He has worked with Oracle Solutions on System z since 2000.

Armelle Cheve is a System z Client Technical Specialist in the Advanced Technical Support (ATS) IBM Oracle Center (IOC), in Montpellier, France. She has 19 years of IT experience, including development in C/C++, SQL, and performance benchmark tests for European customer accounts. She provides presales support to new customers workloads since 2001 for IBM, and then Oracle Solutions on Linux System z. This support includes technical support, consolidation sizings, architecture, and performance studies.

Ivan Dobos is a Project Leader at International Technical Support Organization. He has 15 years of experience with IBM System z. He joined IBM in 2003 and worked in different sales and technical roles supporting mainframe clients as Technical Leader for Linux on System z projects in the System z Benchmark Center, IT Optimization Consultant in the System z New Technology Center and Mainframe Technical Sales Manager in Central and Eastern Europe. During the past 10 years, he worked with many clients and supported new workloads on System z projects.

Damian Gallagher is a Global Technical Lead with Oracle Global Support, working with Oracle solutions on IBM System z. He has worked with databases and specialized in performance for over 25 years.

Hélène Grosch is an IT Architect who has works since 2010 in the IBM Oracle Center in Montpellier, France. She is dedicated to Oracle solutions on Linux on System z projects. Her previous focus was on consolidation projects on Linux on System z solutions in the New Technology Center, Montpellier. She has 15 years of experience in IT. Before joining IBM in 2007, she was a software developer for business applications on mainframes in a consultant and system integration company, mainly for the public sector.

Michael MacIsaac has worked for IBM for 25 years. He is now a z/VM and Linux consultant working in the field with customers on infrastructure and workloads, such as Oracle. He has written technical documentation and presents at conferences and user's group meetings.

Romain Pochard is a System z IT Specialist in the IBM Client Center in Montpellier, France. As part of the Advanced Technical Support organization, he provides technical presales support on Linux on System z and z/VM. His areas of expertise include consolidation, virtualization, and Cloud Computing solutions on IBM zEntreprise.

Barton Robinson is Founder and CTO of Velocity™ Software. After working for IBM in Poughkeepsie in VM Planning, Washington Systems Center for VM Performance, and then in San Jose as a senior manager in the System Performance Evaluation Lab, he started a company to provide VM Performance management products. After 25 years of developing performance software for Velocity Software and providing performance analysis and education, he continues to develop instrumentation and understanding for optimizing Linux performance in a z/VM environment.

David Simpson is a System z Oracle Specialist working for Advanced Technical Support in the Americas Sales and Distribution team. David previously worked for IBM Global Business Services® to host Oracle systems for consulting engagements.

Richard Smrcina is a Senior System Engineer with Velocity Software. He has been in the industry over 25 years, primarily as a z/VM and z/VSE® Systems Programmer. He has installed and supported z/VM and z/VSE his entire career. He was involved with the original IBM Linux for S/390® proof of concept and was instrumental in implementing Linux as one of the first customers to use it in production. With Velocity Software, he provides customer support and is involved in presales, education, customer product installation, and development.

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

- Joe Comitini
- Steve Conger
- Joanne Lazarz
- Giri Krishnan

  ADP

- Dan Davenport

  IBM Dallas

- Andrew Braid
- Sylvain Carta
- Sebastien Llaurency
- Claire Noirault

  IBM France, Montpellier

- Roy Costa

  International Technical Support Organization, Poughkeepsie Center

- Eugene Ong
- Stephen McGarrilll

  IBM STG WW Mainframe Benchmark Center

- David Ong
- Sonia Campins-Punter
- Sandra Skehin

  Oracle

- Terry Elliott
- Thomas Kennelly
- Leon Rich

  IBM USA

- Juergen Doelle

  IBM Germany

Now you can become a published author, too!

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

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

`http://www.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 this website:

  `http://www.ibm.com/redbooks`

- Send your comments in an email to:

  `redbooks@us.ibm.com`

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks
- Follow us on Twitter:

  http://twitter.com/ibmredbooks
- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806
- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm
- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Why customers are choosing to use Oracle products on Linux on IBM System z

Hundreds of users are running their Oracle products on Linux on IBM System z, with more customers finding benefits of the IBM System z platform for deploying Oracle solutions. This chapter reviews some of the top reasons why customers are choosing Linux on System z for their Oracle database deployments. The objective is to help you evaluate your environment to determine whether these reasons might benefit you.

The mainframe was the IT industry’s leading platform for transaction processing, consolidated, secure data serving, and support for enterprise-wide applications. With the IBM zEnterprise® EC12 (zEC12), IBM extends the mainframe platform to help large enterprises reshape their customer experiences through information-centric computing. On a Smarter Planet®, information-centric processes are exploding in growth. To handle vast volumes of data, users need a fast, secure, and available IT infrastructure that helps with the following tasks:

- Improve the user experience with unmatched system security.
- Realize cost savings with 50% increased capacity, so you can invest in new services.
- Use Capacity on Demand options.

The many reasons customers choose Linux on IBM System z for their mission-critical applications or data servers can vary. The reasons are based on the unique advanced architecture of the IBM System z infrastructure, which provides many high-availability advantages and unique characteristics. Your IBM representative can provide more detailed information.

This chapter includes the following topics:

- Virtualization capabilities of IBM System z
- Ability to use existing disaster recovery plans
- Trusted Security and Resiliency
- System z is optimized for High Availability
- Total cost of ownership advantages of IBM System z
- Ease of interfacing with traditional data
- Increased performance and scalability capabilities of System z, including zEC12, z114, and z196
- Specialty engines available on IBM System z
- IBM zEnterprise BladeCenter Extension
- End-to-end solution for dynamic infrastructure data center
- Cost savings
- Ability to easily add more capacity
- IBM Cloud Services
- Summary
- Oracle solutions available on IBM System z

1.1 Virtualization capabilities of IBM System z

The advanced and superior virtualization capabilities of z/VM and System z are a key reason why running Oracle on System z is a popular solution.

z/VM provides highly flexible production and test environments for enterprises that are deploying database and business solutions.

z/VM helps enterprises meet their growing demands for multi-system server solutions with a broad range of support for operating system environments, including z/OS® and Linux on System z.

z/VM offers the following benefits:

- Mature technology:
  - VM/370, introduced in 1972, VM/370 was enhanced and significantly expanded for Linux in recent years.
  - Infrastructure support:
    - Back up and Disaster Recovery
    - Performance Management
- Software Hypervisor integrated with hardware:
  - Effective sharing of CPU, memory, and I/O resources
  - Virtual network - virtual switches and routers
  - Virtual I/O (mini-disks, virtual cache, and so on)
  - Virtual appliances
- Easy of management:
  - Fast cloning or provisioning of preinstalled and configured Linux images. This can be accomplished in minutes instead of days or weeks.
  - Reduced space, electric connections, or network cables.
  - Compatible with the data center practice of standardizing on strategic software stacks with consistent levels and patches.

The capability to rapidly deploy Linux guests with Oracle databases is used by many customers in their infrastructure simplification strategy.

Tip: For more information about the benefits of virtualization on System z, see *Using IBM Virtualization to Manage Cost and Efficiency*, REDP-4527-00, at this website:

http://www.redbooks.ibm.com/abstracts/redp4527.html

1.2 Ability to use existing disaster recovery plans

For most customers that are running on System z, there are well-established business processes and disciplines for disaster recovery and business resiliency already in place in their organization. Oracle solutions on the mainframe also can easily fit into the disaster recovery infrastructure that is already in place for the mainframe.

This reason was highlighted by many users as one of the key reasons they moved their Oracle databases to Linux on System z.

1.3 Trusted Security and Resiliency

IBM System z is a leading platform for secure data serving and the only commercial server to achieve Common Criteria Evaluation Assurance Level 5+ security classification, which provides clients the confidence to run many different applications that contain confidential data on a single mainframe. The new zEC12 builds on this with innovative security and privacy features to help protect data at rest or in flight, which is a critical capability in the age of Internet banking and mobile devices.

The zEC12 includes a state-of-the-art, tamper-resistant cryptographic adapter, Crypto Express4S that provides privacy for transactions and sensitive data. Crypto Express4S includes new hardware and firmware that was developed with IBM Research to help meet the security requirements of different industries and geographies. For example, it can be configured to provide support for high-quality signatures that are used with applications for Smart passports, national ID cards and online legal proceedings, and meet standardized cryptographic services and certifications that are needed by the European Union and public sector.

1.4 System z is optimized for High Availability

System z is a large-scale consolidation system that thrives under heavy workloads. There are clients that deploy hundreds of thousands of millions of instructions per second (MIPS), often while they are running thousands of programs, hundreds of applications, massive amounts of data, and huge numbers of users with long periods of uninterrupted operation. System z continues to build on decades of design of hardware and software to keep the system up and running to provide the highest availability and is unmatched with 99.999% reliability. IBM added technology innovations to further extend that availability that helps improve handling of paging workload spikes and IBM zAware to increase availability as a self-learning integrated analytics solution.

For more information, see Chapter 9, “High Availability and Disaster Recovery environment for Oracle” on page 179.

1.5 Total cost of ownership advantages of IBM System z

When you are looking at the total cost of ownership (TCO) for an Oracle on Linux on System z implementation, many users find the lowest TCO with the System z solution. The costs savings come from a combination of factors, including hardware, software, operations, and management costs. IBM has tools available that can assist customers in determining their TCOs for a planned project, including RACEv, Eagle, and Scorpion.

The ability to run multiple images of Oracle on fewer processors with System z and Linux where processor usage levels of 85% or more can be achieved brings more benefits in the area of licensing and hardware costs. Associated with this are the “green” features of System z that bring other advantages of reduced power requirements and reduced floor space requirements, as shown in Figure 1-1 on page 5.

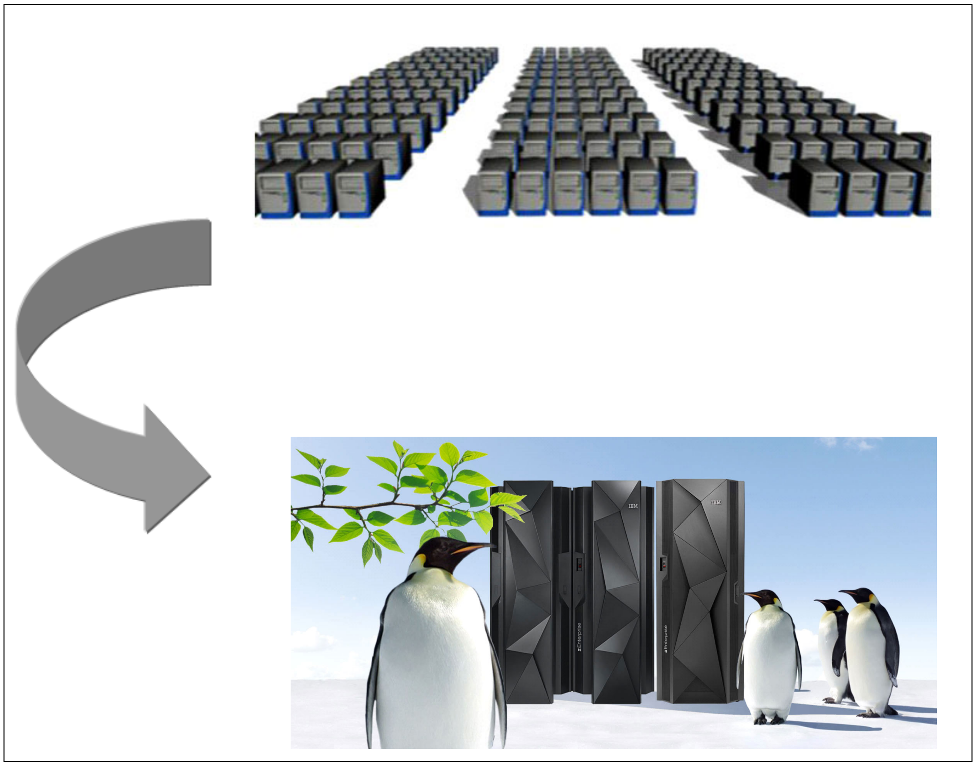


Figure 1-1 Server Farm to a Lean Green data center

By using z/VM, you can create a virtual machine for short-term Oracle database use. For example, if a DBA must resolve a unique problem or handle a specific test and then recycle the resources back to the pool when completed rather than installing and uninstalling a physical box.

The mainframe includes the following “green” features:

- Distributed servers (including production servers, development servers, test servers) often run at average usage levels of 5% - 20%.
- Virtualization and workload management enable standardization and consolidation on the mainframe.
  - Run multiple images on fewer processors
  - Achieve usage levels of 85% or more without performance degradation
- Become “lean and green” through IT consolidation and infrastructure simplification, as shown in Figure 1-1.

1.6 Ease of interfacing with traditional data

For over 40 years, many companies’ mission-critical data was stored in the mainframe. By running Oracle solutions on Linux on System z, the interface to this data is handled easily and efficiently. The use of HiperSockets™ enables speedy access to this data.

1.7 Increased performance and scalability capabilities of System z, including zEC12, z114, and z196

System z continues to grow and deliver increased performance for applications that need more performance capability. The IBM zEnterprise EC12 (zEC12) was designed to help users meet today's extreme data serving demands with three key capabilities: trusted security and resilience, efficiency at scale, and operational analytics. The zEC12 enterprise system was the result of an investment by IBM Systems and Technology Group of more than $1 billion in IBM research and development (primarily in Poughkeepsie, New York) and 17 other IBM labs around the world in collaboration with some of IBM's top clients.

The next generation IBM zEnterprise System offers a proven hybrid computing design that can manage workloads on multiple platforms with the simplicity of a single system. New 5.5 GHz processors deliver increased performance and cross platform virtualization solutions to help address server sprawl. The industry's premier enterprise infrastructure supports today's smarter computing solutions. Also, new options for physical planning, such as non-raised floor and overhead I/O and power, provide increased flexibility for disaster recovery solutions.

1.8 Specialty engines available on IBM System z

Different specialty engines that are available on IBM System z. One of these engines, Integrated facility for Linux (IFL), plays a key role in Linux on System z environments

The following cost savings are significant based on IFLs:

- IFL processors for Linux are more engines that are dedicated to Linux workloads:
  - Support z/VM and Linux on System z
  - IFL processors run at "full speed" (5.5 GHz on zEC12)

    zEC12, z196, z114, z10™ EC, z10 BC
- Traditional IBM and independent software vendor (ISV) mainframe software charges are unaffected.
- Linux and z/VM charged only against the IFLs.
- Subcapacity pricing for Oracle available through hard partitions with IFLs on System z and LPARs

1.9 IBM zEnterprise BladeCenter Extension

Being heterogeneous makes sense if there are no issues with management and integration. The IBM zEnterprise System offers a proven hybrid computing design that can help you manage and integrate workloads on multiple architectures with the simplicity of a single system. The zEC12 supports heterogeneous platform requirements with the new IBM zEnterprise BladeCenter® Extension (zBX) Model 003 and IBM zEnterprise Unified Resource Manager (zManager) for extending management strengths to other systems and workloads running on AIX® on POWER7®, Linux on IBM System x® and Microsoft Windows on IBM System x servers.

zBX is the new infrastructure for extending System z governance and management capabilities across a set of integrated, fit-for-purpose POWER7, and IBM System x compute elements in the zEnterprise System. This expands the zEnterprise portfolio to applications that are running on AIX, Linux on System x, and Microsoft Windows. Attachment of the zBX to the zEnterprise Central Processing Complex (CPC) is via a secure high-performance private network. The blades can help increase flexibility in “fit for purpose” application deployment.

1.10 End-to-end solution for dynamic infrastructure data center

The dynamic infrastructure data center is the data center of the future. This is driven by the need for greater energy efficiency, which is a global issue with significant impact today with an even greater impact in the future.

Data center design must change. Technology and business growth uncertainty and rising costs drive the need for the following new approaches:

- IT efficiency enables energy efficiency.
- IT and facilities must work together.

Energy efficiency is a key metric to evaluate overall IT operational efficiency.

The dynamic infrastructure data center is an evolutionary new model for efficient IT delivery that includes the following aspects:

- New economics: Virtualization with optimized systems and networks to break the lock between IT resources and business services.
- Rapid service delivery: Service management enables visibility, control, and automation to deliver quality service at any scale.
- Aligned with business goals: Real-time integration of transactions, information, analytics, and delivery of IT as a service.

1.11 Cost savings

All the issues described thus far in this chapter contribute to the cost savings that are achieved by running Oracle solutions on the mainframe.

Potential savings include the following examples:

- Deploying virtual servers can reduce hardware requirements, which can result in savings when purchasing, installing, and configuring new hardware. Running servers at higher usage means that fewer servers are required.
- Fewer hardware servers occupy less space, which can result in savings on raised floor requirements, heating, cooling, and electricity.
- Virtual servers can be created in minutes, which can help reduce cost and time that is associated with planning for new business requirements.
- Sharing operating systems and application code between virtual servers can save on software costs, operating costs, Systems Management, and staffing.
- System management tools are delivered as part of the system, which can help avoid the cost of more software to perform these tasks.

- Network costs can be reduced because virtual servers communicate by using HiperSockets or z/VM Guest LANs.
- Oracle's multi-core pricing applies to IBM System z, including zEC12, z196, z114, z10 EC, and z10 BC.

Because of these savings, Oracle on System z often offers the best TCO with the best service level.

1.12 Ability to easily add more capacity

By using the following options, you can increase capacity:

- Capacity Upgrade on Demand (CUoD)

  CUoD provides the capability to permanently add CPs, ICFs, IFLs, zAAPs, zIIPs, SAPs, memory, and channels non-disruptively, which eliminates the need for a scheduled outage. Installations that make use of the CUoD option can use the added capacity non-disruptively. If power-saving mode is turned on, the CUoD function is blocked.
- On/Off Capacity on Demand (On/Off CoD)

  When your business needs short term additional capacity, On/Off CoD is designed to deliver it. On/Off CoD is designed to temporarily turn on CPs, IFLs, ICFs, zAAPs, and SAPs. Up to eight temporary records (CBU, CPE, and On/Off CoD) can be installed and activated at any time. You also have the flexibility of activating some of the resources on a record. You do not have to activate the entire record. You also can add capacity and engines and extend the duration of the temporary upgrade concurrently, which eliminates the need for constant ordering of new temporary upgrade.
- Capacity Backup (CBU)

  The CBU capability (temporary upgrade) enables enterprises to provide flexible, cost-effective Disaster Recovery on System z.
- Capacity for Planned Events (CPE)

  CPE is designed to replace lost capacity within a customers' enterprise for planned downtime events, such as system migration or relocation (for a data center move). This temporary upgrade is available for three days.

Software licensing: It is your responsibility to ensure that your Oracle software license agreement covers the other temporary or permanent capacity.

1.13 IBM Cloud Services

With faster processing and extensive capacity, System z is positioned as a cloud server that offers a high degree of efficiency and the ability to scale into large cloud configurations.

1.14 Summary

IBM System z brings the following advantages to Oracle and Linux:

- The most reliable hardware platform available:
  - Redundant processors and memory
  - Error detection and correction
  - Remote Support Facility (RSF)
  - Non-disruptive hardware updates
  - Highest level of virtualization available
- Centralized Linux systems are easier to manage.
- Designed to support mixed workloads:
  - Allows consolidation while maintaining one server per application
  - Complete workload isolation
  - High-speed inter-server connectivity
- Massively scale your workload on a single System z mainframe:
  - Host many Linux virtual machines on z/VM.
  - Each virtual machine on z/VM can access up to 24,576 devices.
  - System zEC12 scales to 101 configurable cores.
  - zEnterprise 196 scales to 80 configurable cores.
  - zEC12's microprocessor speed of 5.5 GHz, the fastest superscalar processor in the industry.
  - More on-chip cache to speed data serving to the processors.
- IBM System z mainframe:
  - Legendary dependability
  - Extremely security-rich:
    - Highest security classification for general-purpose servers.
    - System z LPAR technology is EAL 5 certified.
  - Designed for multiple diverse workloads running concurrently.
  - Proven high volume data acquisition and management.
  - Options to easily increase capacity.
  - HiperSockets provides high-speed communications between servers:
    - Private interconnect via HiperSockets reduces cache fusion latency significantly.
    - Connecting between batch servers and database servers via HiperSockets contributes to reducing elapsed time.
- System z offers the ultimate in virtualization with z/VM, which virtualizes everything with high levels of usage:
  - CPU, memory, network, I/O, cryptographic features, Coupling Facility
  - Support for large real memory
  - Enhanced security and LDAP server/client
  - Enhanced memory management for Linux guests
  - Enhanced management functions for Linux

- The open standards operating system of Linux on System z:
  - Reliable, stable, security-rich
  - Available from multiple distributors
  - Plentiful availability of skilled administrators and developers
  - Large selection of applications, middleware, and tools from IBM, ISVs, and open source

1.15 Oracle solutions available on IBM System z

Oracle has the following families of products available on IBM System z:

- Oracle Database Server
- Oracle Fusion Middleware
- Oracle Application Suites (DB Tier runs on z):
  - Oracle E-Business Suite
  - Oracle's PeopleSoft Enterprise
  - Oracle's Siebel applications

The availability of Oracle products on each Linux distribution is constantly changing. For more information, see this website:

`http://support.oracle.com`

Part 1

Setting up and installing Oracle 11gR2 on Linux on System z

In the first part of this book, we describe the following processes:

- How to get started with a project that is running the Oracle Database on Linux on System z in Chapter 2, "Getting started on a proof of concept project for Oracle Database on Linux on System z" on page 13.
- How to set up the Network Infrastructure for Grid Control in Chapter 3, "Network connectivity options for Oracle on Linux on IBM System z" on page 29.
- How to install and set up a Red Hat Enterprise Linux 6 and SUSE Linux Enterprise Server 11 guest for Oracle in Chapter 4, "Setting up SUSE Linux Enterprise Server 11 SP2 and Red Hat Enterprise Linux 6.2" on page 55.

  For more information, see Appendix A, "Setting up Red Hat Enterprise Linux 6.3 for Oracle" on page 311 and Appendix B, "Installing Oracle and creating a database 11.2.0.3 on Red Hat Enterprise Linux 6" on page 335.
- How to install the Grid Control Agent in Chapter 5, "Using the Cloud Control agent to manage Oracle databases" on page 71.

The objective is to provide you with the current best practice information for setting up Linux for Oracle, get the Oracle Database binaries installed, and create an Oracle single instance database.

For more information about installing Real Application clusters, see the following publications:

- *Silent Installation Experiences with Oracle Database 11gR2 Real Application Clusters on Linux on System z,* REDP-9131
- *Installing Oracle 11gR2 RAC on Linux on System z,* REDP-4788

2

Getting started on a proof of concept project for Oracle Database on Linux on System z

This chapter describes the important choices that you must consider during the planning process of setting up a project for an Oracle 11gR2 Database.

The following key concepts are important:

- A proper Integrated facility for Linux (IFL) and memory sizing that is done in advance is necessary to understand the capacity requirements for the proof of concept (PoC).
- The virtualization benefits on System z fit well with Oracle Database technologies, which provide many architecture alternatives to match business needs.
- A high-quality disk infrastructure is key to a successful outcome.
- PoCs with smaller and less important Oracle Databases make better candidates for the initial PoC.
- Oracle Databases that are implemented on Linux on System z can handle Enterprise Resource Planning (ERP)-sized databases and the following servers:
  - PeopleSoft database server
  - Siebel database server
  - E-Business suite database server
- Oracle Database on Linux on System z can handle data warehouse applications.

IBM and Oracle continue to work together and invest in improving the Oracle on Linux on System z solutions.

2.1 Architecture

There are several options for how you can configure your environment to run an Oracle Database on Linux on System z. The choice depends on the requirements for High Availability (HA) and disaster recovery. The following choices are described in this chapter:

- Single instance, including Automated Storage Manager (ASM)
- Real Application Cluster (RAC) One node
- Multinode RAC on one logical partition (LPAR)
- Multinode RAC on more than one LPAR on one Central Processor Complex (CPC)
- Multinode RAC on multiple CPCs
- Use of Data Guard
- Use of GoldenGate

Tip: Figures in this chapter show the architecture under z/VM. You also have the option of running Linux native on the IFL with no z/VM virtualization.

2.1.1 Single Instance database

The Single Instance database is the simplest setup. The IBM System z provides strong hardware availability. This is the most frequently used option for smaller databases. Although Figure 2-1 is overly simplistic, it is expanded as the different availability options are described.

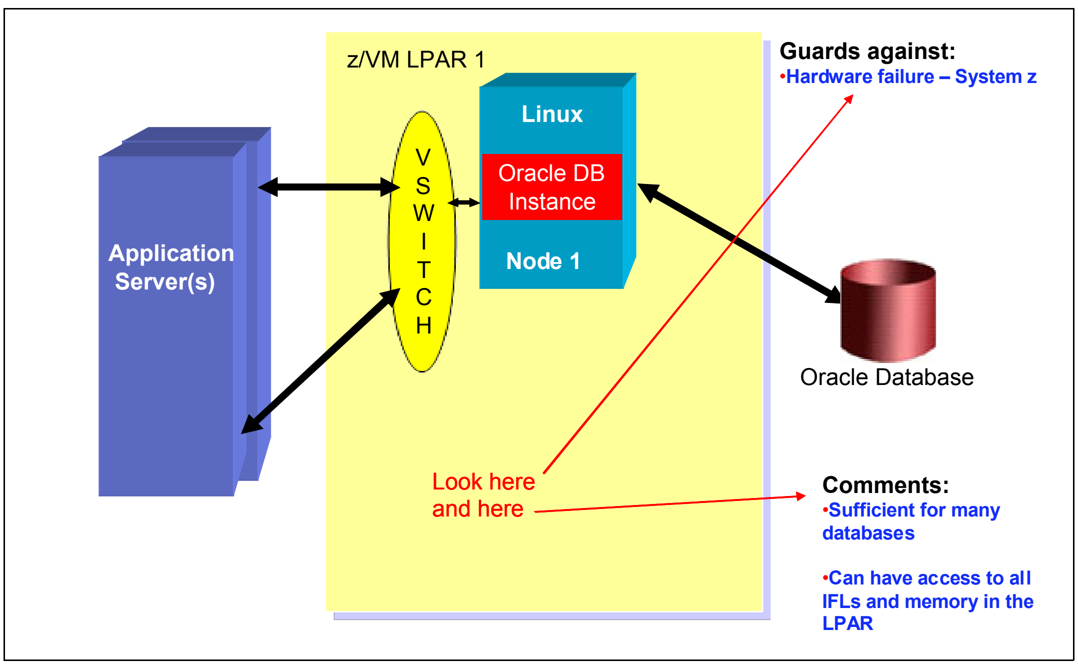


Figure 2-1 Single Instance database

You can add to this set up by using Oracle's ASM as shown in Figure 2-2.

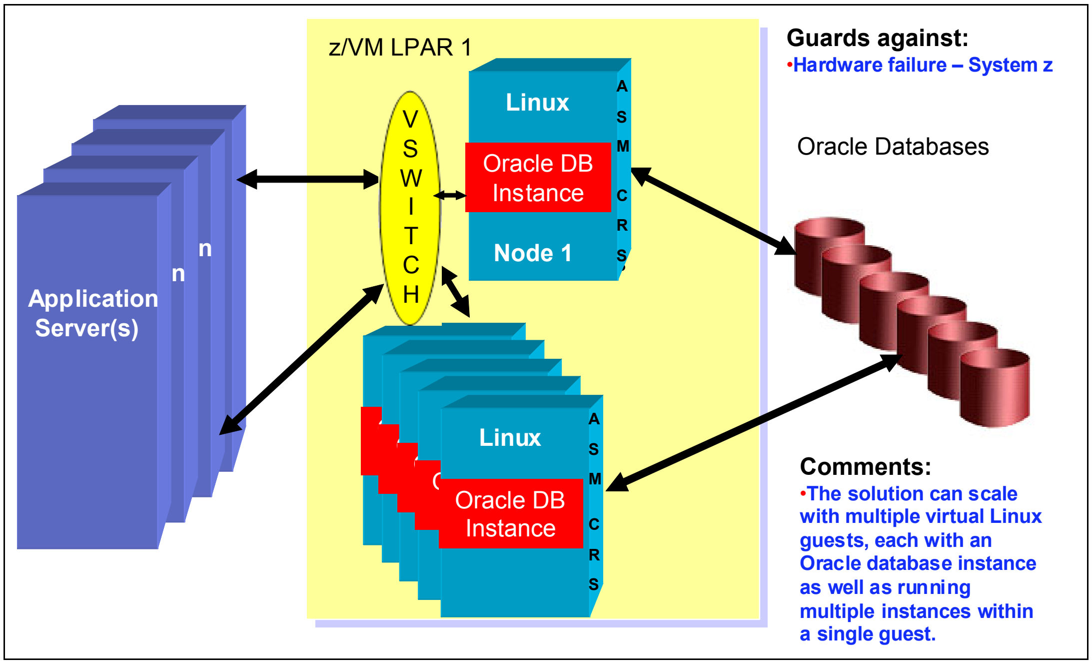


Figure 2-2 Oracle Databases virtualized by using z/VM on System z

2.1.2 Single Instance with Cluster Ready Services or RAC One-Node

Oracle offers a one node RAC option that provides a failover option. This is a popular option that provides some failover capabilities, as shown in Figure 2-3.

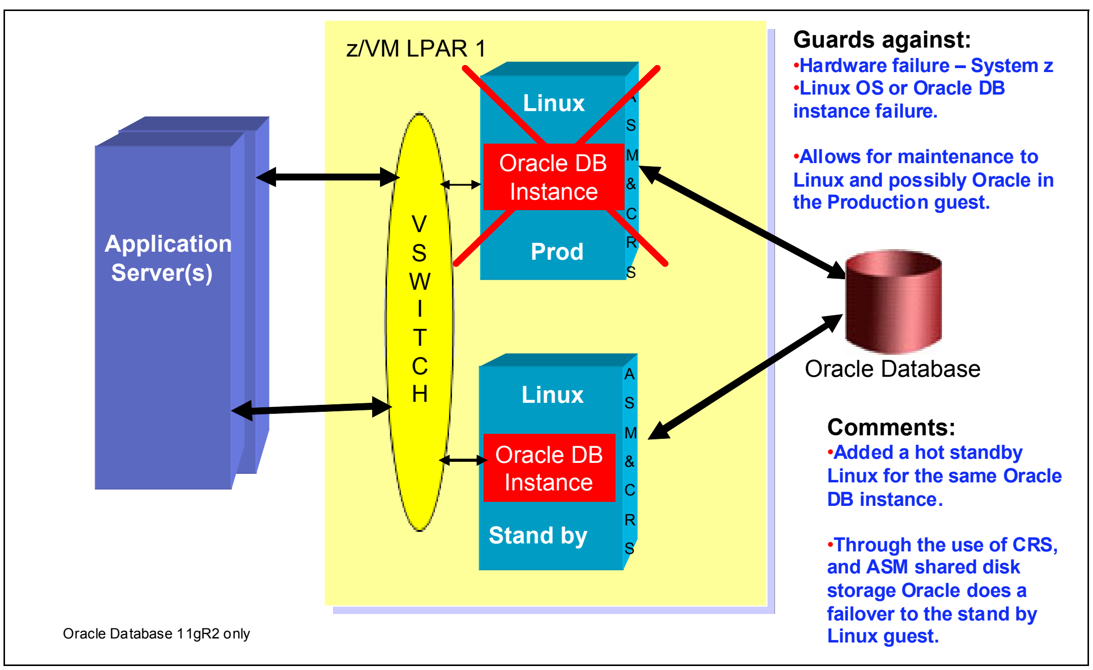


Figure 2-3 RAC One Node

This RAC One Node option also can be accomplished across LPARs that use HiperSockets connections in the same CPC. It can also be accomplished across different System z systems by using appropriate network connectivity. It is only allowed between Oracle Databases that use the same binaries (in this case, Linux on System z) at the same level.

Oracle CRS and Oracle RAC do not support heterogeneous platforms in the same cluster. For example, you cannot have one node in the cluster that is running Linux on System z and another node in the same cluster that is running Solaris UNIX. All nodes must run the same operating environment; that is, they must be binary compatible. Oracle RAC does not support machines that have different chip architectures in the same cluster. However, you can have machines of different speeds and sizes in the same cluster.

RAC One Node's availability option is less capable than that of a full RAC implementation. Therefore, outages can be a short duration with this option. Some database recovery might need to occur in the failover node.

One advantage of System z in this scenario is that the failover guest uses few IFL and memory resources until they are needed for an actual failover. After the failover occurs, IFL and memory resources from production guest become available to the failover guest, which negates the need for significant duplicate IFL and memory resources that are required in other hardware architectures.

2.1.3 Two-node RAC on the same LPAR

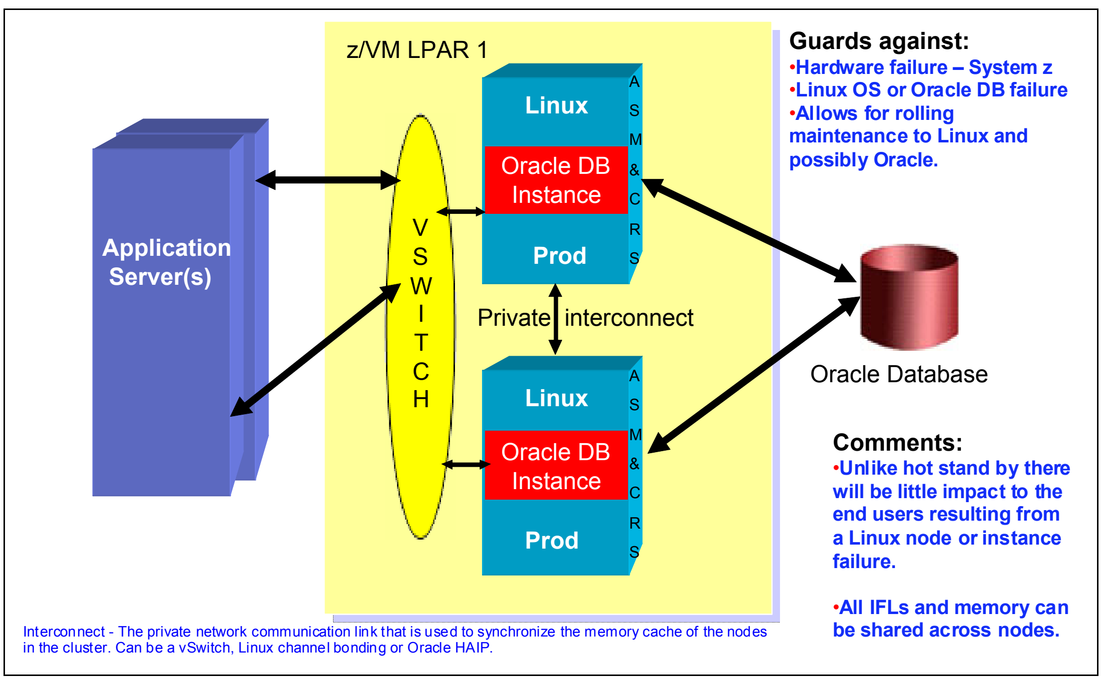


Figure 2-4 Multinode RAC on one LPAR

RACs can be used in the following ways:

- Active/Passive configuration

  One node processes work while the other node waits for the first node to fail.

- Active/Active configuration

  All nodes process work; if any node fails, the cluster is remastered.

In addition to availability, RAC can be used for workload distribution because all work does not have to go through all nodes. Oracle RAC can be deployed in the following ways:

- In the same LPAR for test and development applications
- Across LPARs for LPAR maintenance or software failures (most common implementation)
- Across CPCs when taking entire systems down is a common occurrence

These options are shown in Figure 2-4 on page 16, Figure 2-5, and Figure 2-6 on page 18.

2.1.4 Multinode RAC on more than one LPAR on one CPC

The multinode RAC on several LPARs on the same CPC is shown in Figure 2-5.

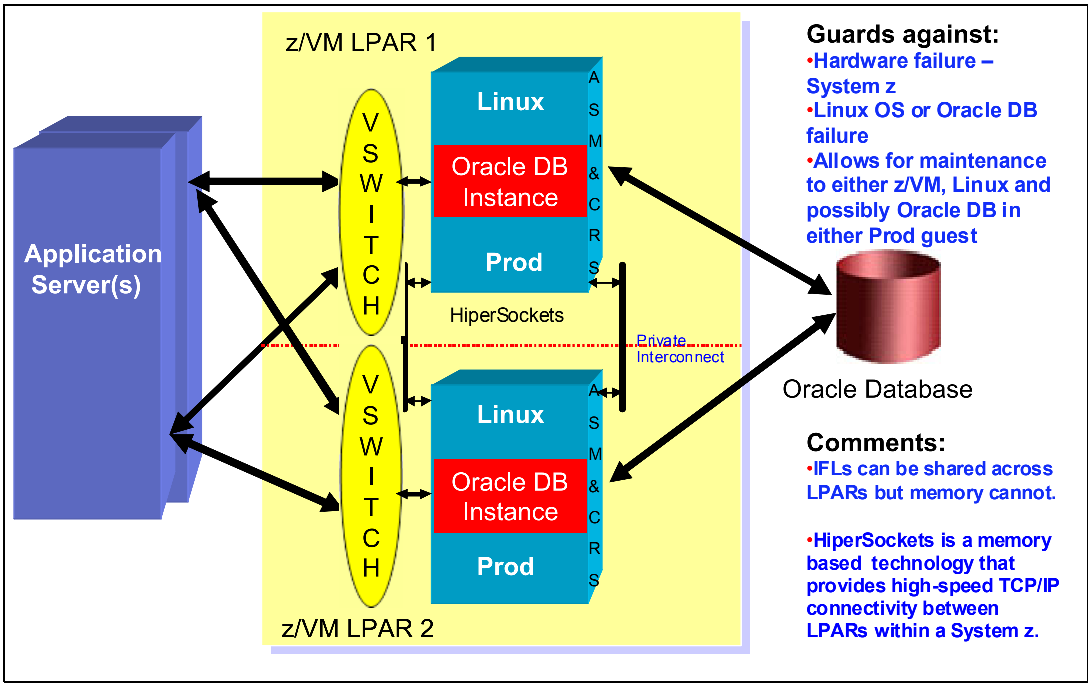


Figure 2-5 Multinode RAC on two or more LPARs

2.1.5 Multinode RAC in two or more CPCs

Figure 2-6 shows the multinode RAC through several CPCs scenario.

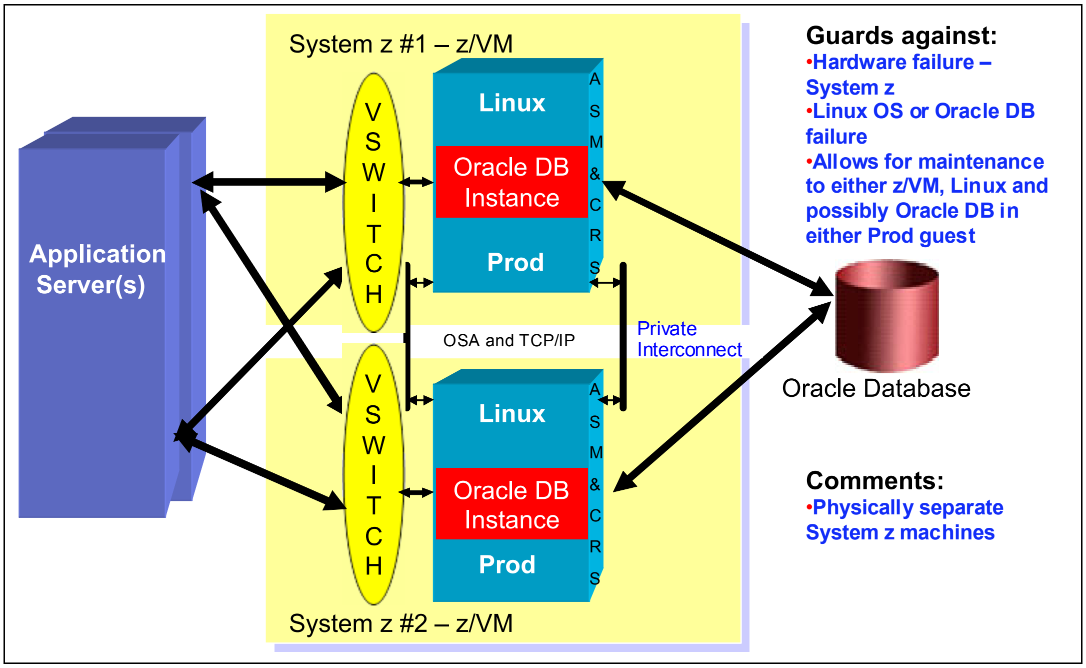


Figure 2-6 Multi-node RAC on two or more CPCs

2.1.6 Data Guard

For disaster recovery scenarios, you might consider Oracle Data Guard if you are focused solely on Oracle products, as shown in Figure 2-7.

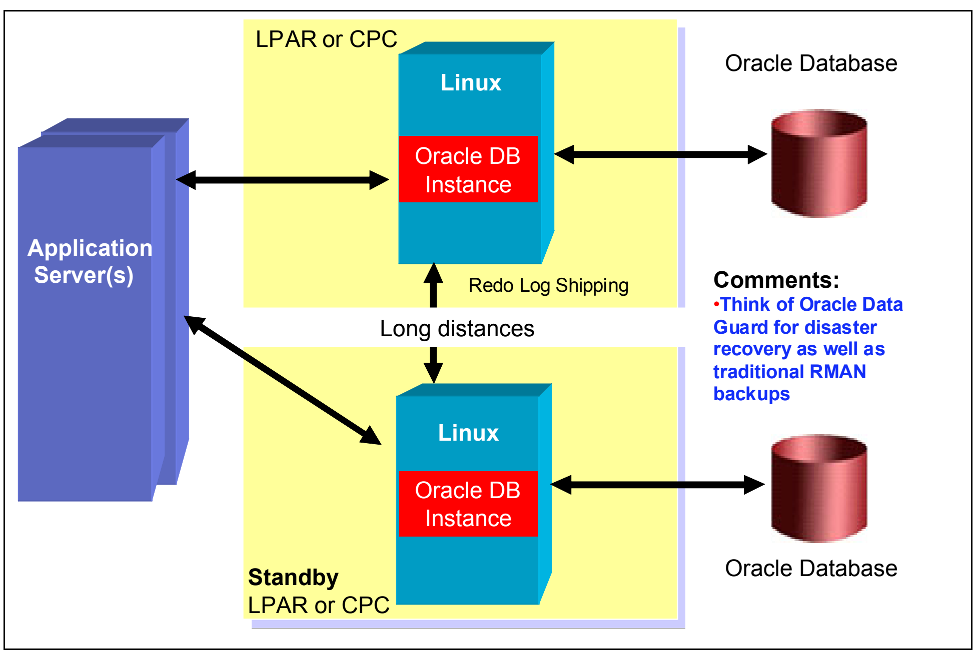


Figure 2-7 Using Oracle Data Guard

As shown in Figure 2-7 on page 18, Standby is the replication to the standby database.

Oracle Data Guard replicates the database to the standby database and includes the following features:

- Uses redo log shipping for log apply or SQL apply
- Less data transmitted than replication
- Sync or async modes are available
- Various configurations of logical and physical standby databases

Important: Support for Data Guard on heterogeneous systems is not certified; therefore, the primary system and the standby system must match for endian formats, chip sets, and headers.

Data Guard often is deployed between CPCs.

2.1.7 Using GoldenGate for replication

Oracle's GoldenGate can also be used as a disaster recovery solution, but it has different capabilities than Data Guard. GoldenGate can move data across heterogeneous databases whereas Data Guard cannot.

2.1.8 Summary

System z is recognized as a highly available platform. This is based on the attention to detail over decades of engineering. It has a fault tolerant (HA) design that is based on the elimination of single points of failure.

Oracle Database HA options blend well with IBM System z to provide a highly available solution. The synergies of System z HA design can be augmented with the various levels of Oracle Database HA to achieve the required level of availability. Additionally, each guest that is running Oracle Database (see Figure 2-2 on page 15), can be customized to meet those levels of availability instead of requiring a "one size fits all" approach.

2.2 Sizing

The sizing process is the most important step in the planning stage for a successful PoC or a production implementation.

This section describes the sizing tool options that are available and the sources of vital input data.

The objective is to plan for the correct number of IFLs and the memory that is needed for the Linux guest to run the Oracle Database.

2.2.1 Sizing tool

There are several choices of tools that can help you in the sizing process, as shown in Figure 2-8. The one most commonly used tool for Oracle Database sizing on IBM System z is the Server Consolidation (SCON) tool with the SURF option. These are IBM internal tools that take customer input and result in an IFL sizing.

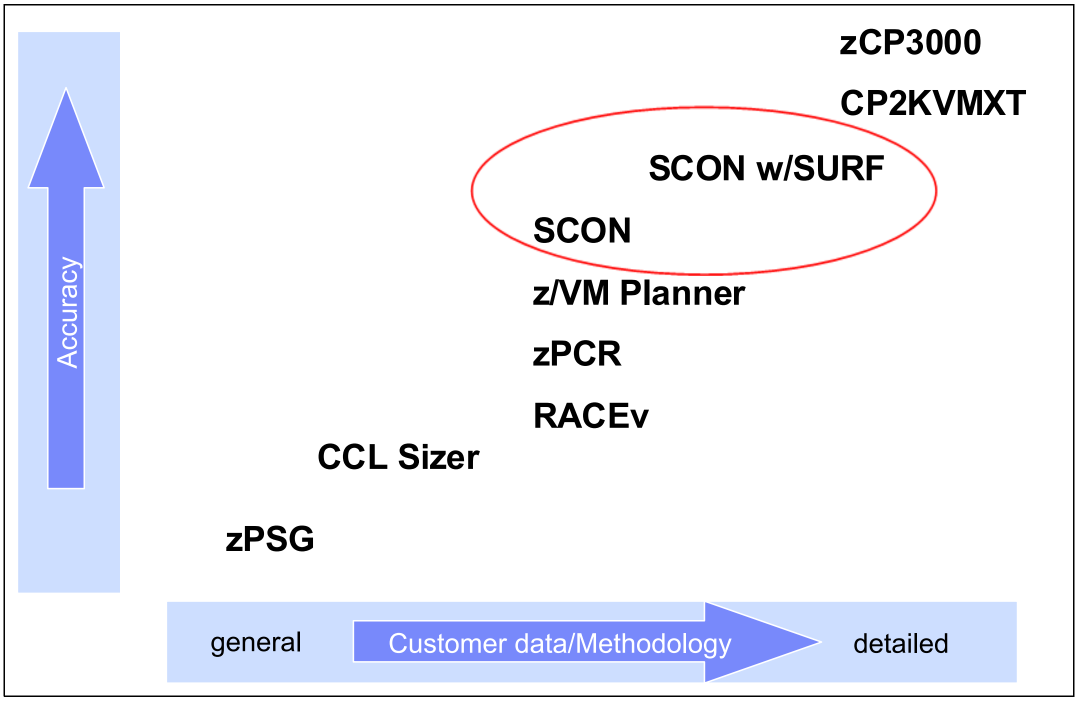


Figure 2-8 Sizing tools for System z

The SCON tool includes the following steps:

1. Complete the IBM provided questionnaire for all your distributed servers that are considered for server consolidation onto System z.
2. Submit the questionnaires to IBM Techline or an IBM sizing specialist who enters the data into a tool to produce the report that is shown in the upper right corner of Figure 2-9 on page 21.
3. If you measured data for the SURF tool, IBM combines the measured data within the SCON tool for a more comprehensive analysis.
4. The SCON tool produces a report that is shown in the lower right of Figure 2-9 on page 21 that indicates the IFL capacity that is required to run your Oracle Database (or databases).

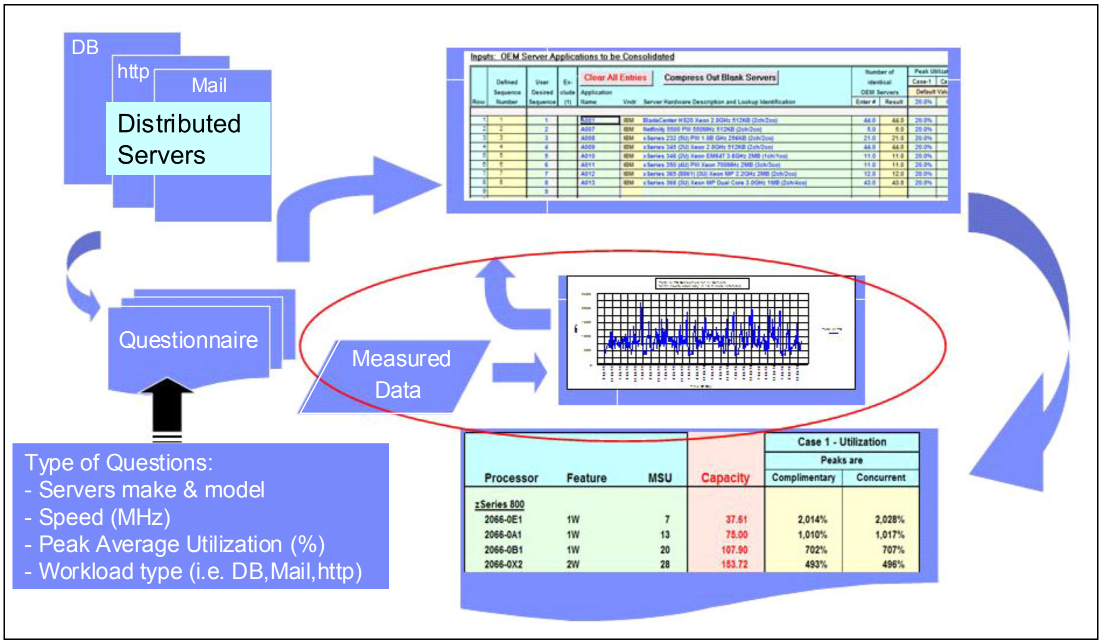


Figure 2-9 Process for sizing with SCON tool and SURF report

2.2.2 Memory sizing

To understand the amount of memory that is required in the consolidated environment, you can perform the following tasks:

- Obtain Oracle System Global Area (SGA) and Program Global Area (PGA) sizes from all database instances.
- Use Oracle SGA and PGA memory advisory sizes from multiple AWR reports, as shown in Figure 2-10 on page 22 and Figure 2-11 on page 22.
- Calculate individual guest storage requirements (assuming MB):

```
Sum of (optimized) SGA and PGA settings +
256 MB for ASM (if used) +
512 MB for Linux +
512 MB for Oracle Enterprise Manager 12c agent (if used) +
dedicated threads memory +
10%[1]
```

- Apply a z/VM memory over commit factor such as 1.2 or 1.5 for production and potentially a higher over commit for test and development environments.

System z memory is the sum of real memory for guests as calculated previously and the memory for z/VM and expanded storage.

[1] Increase estimate when Oracle SGA is large and there are expected to be hundreds of dedicated server connections or use Linux hugepages with Oracle 11gR2.

PGA Memory Advisory

- When using Auto Memory Mgmt, minimally choose a pga_aggregate_target value where Estd PGA Overalloc Count is 0

PGA Target Est (MB)	Size Factr	W/A MB Processed	Estd Extra W/A MB Read/ Written to Disk	Estd PGA Cache Hit %	Estd PGA Overalloc Count
896	0.13	148,138.91	182,994.64	45.00	1,297
1,792	0.25	148,138.91	173,054.91	46.00	1,197
3,584	0.50	148,138.91	30,487.16	83.00	0
5,376	0.75	148,138.91	30,487.16	83.00	0
7,168	1.00	148,138.91	29,701.39	83.00	0
8,602	1.20	148,138.91	12,032.42	92.00	0
10,035	1.40	148,138.91	12,032.42	92.00	0
11,469	1.60	148,138.91	12,032.42	92.00	0
12,902	1.80	148,138.91	12,032.42	92.00	0
14,336	2.00	148,138.91	12,032.42	92.00	0
21,504	3.00	148,138.91	12,032.42	92.00	0
28,672	4.00	148,138.91	12,032.42	92.00	0
43,008	6.00	148,138.91	12,032.42	92.00	0
57,344	8.00	148,138.91	12,032.42	92.00	0

Figure 2-10 Using PGA Advisory

Using the AWR report for the SGA size

A memory size of 4 GB - 9 GB gives a reasonable trade-off between SGA memory allocation and reduction in physical reads, as shown in Figure 2-11 and Figure 2-12 on page 23. In this case, 9 GB was used.

SGA Target Advisory

SGA Target Size (M)	SGA Size Factor	Est DB Time (s)	Est Physical Reads
2,304	0.25	317,428	14,503,025
4,608	0.50	282,694	11,631,530
6,912	0.75	270,413	10,965,119
9,216	1.00	263,535	10,396,434
11,520	1.25	258,791	10,003,449
13,824	1.50	255,418	9,725,864
16,128	1.75	252,915	9,517,935
18,432	2.00	252,150	9,454,517

Figure 2-11 SGA Target Advisory

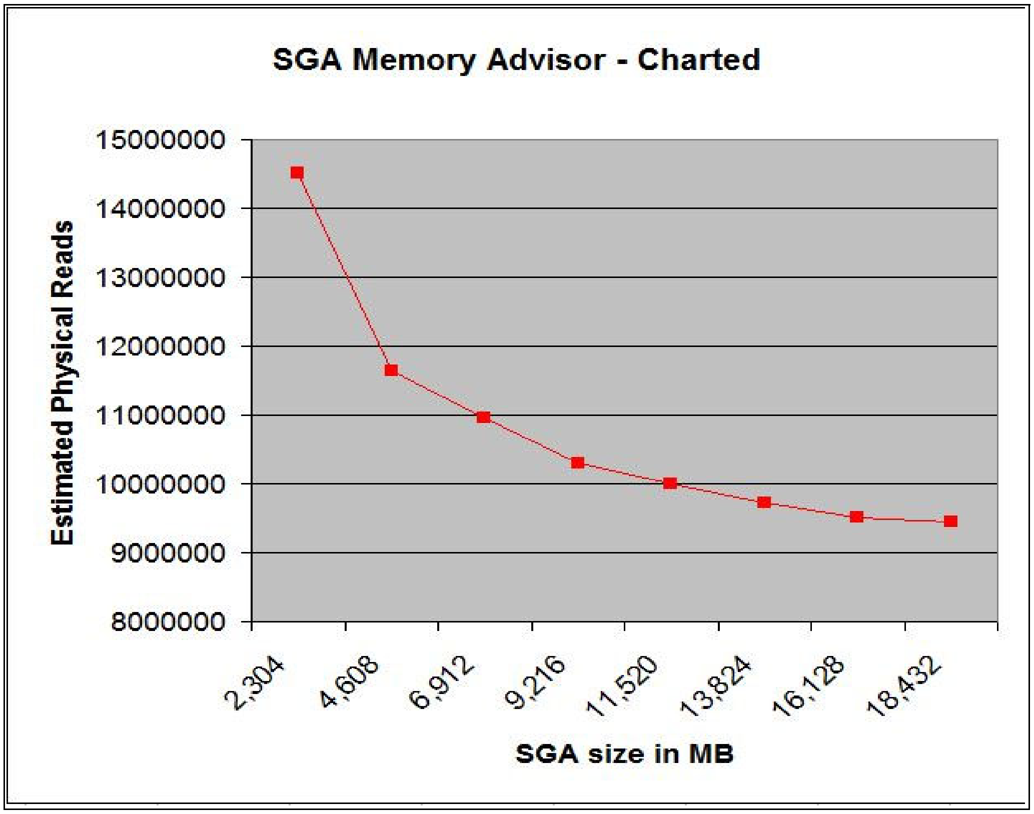
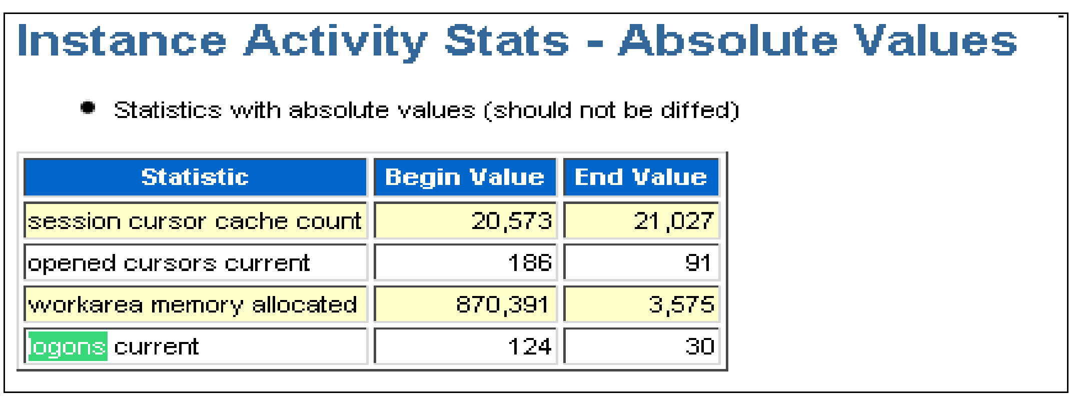


Figure 2-12 SGA Memory Advisor

2.2.3 Threads for dedicated processes

To size for the memory for dedicated threads, you should determine the number of dedicated threads and multiply that number by 4.5 MB to determine the required real memory to include in guest sizing.

Information from the Automatic Workload Repository (AWR) report can provide some additional information. The logon information as shown in Figure 2-13 might provide a hint about the number of threads in use, if dedicated.



Instance Activity Stats - Absolute Values

- Statistics with absolute values (should not be diffed)

Statistic	Begin Value	End Value
session cursor cache count	20,573	21,027
opened cursors current	186	91
workarea memory allocated	870,391	3,575
logons current	124	30

Figure 2-13 Number of log ons

2.2.4 Summary

Remember the following key points about sizing:

- Use the most accurate data that is available. Take the time to look at the Oracle AWR reports. The "garbage in, garbage out" rule applies here. Do not make assumptions about or guess the input to the sizing tool.
- Choose the most appropriate time of day or month to measure the existing load so you are working with meaningful data.
- After you complete the sizing process and are ready to implement it, ensure that the CPU (IFL) capacity and the memory that is required is available, and have a high-quality I/O infrastructure in place.
- Contact an IBM System z Oracle Specialist to help with the sizing process or request sizing assistance from IBM Techline.

2.3 I/O considerations

Of equal importance to proper sizing is an effective I/O subsystem infrastructure for your Oracle Databases on Linux on System z.

The following generic disk infrastructure is available:

- DASD (ECKD™ - 3390) or SCSI
- Oracle's ASM or Logical Volume Manager (LVM) with stripping
- Type of disk for z/VM, Linux, and Oracle binary code
- Type of disk for the Oracle Database:
  - SCSI/FCP
  - DASD

The following infrastructure alternatives give the best database performance:

- HyperPAV subsystem on DASD and HyperPAV driver support in Linux distribution
- Use of SSD
- IBM XIV®
- Level of the Linux code:
  - Red Hat Enterprise Linux 6.x or 5.9
  - SUSE Linux Enterprise Server 11 SP 2

2.3.1 Fibre Channel Protocol

When you are using SCSI disks, you must use the following Fibre Channel Protocol (FCP) channels on the System z:

- FCP CHPID planning considerations
- Defining logical unit numbers (LUNs) (scanning)
- Multipath considerations (permissions and performance)
- Recommend creating UDEV rules when Oracle's ASM is used
- Implement N-Port ID Virtualization (NPIV) for individual guest disk security, as required

2.3.2 ECKD and DASD

When you are using ECKD DASDs, the following considerations should be reviewed:

- HyperPAV/PAV setup
- Configuring UDEV rules for DASD

For more information, see the following publications:

- *My Oracle Support note 1351746.1 How to Manually Configure Disk Storage devices for use with Oracle ASM 11.2 on IBM: Linux on System z under RedHat 5*.
- *My Oracle Support note 1350008.1 How to Manually Configure Disk Storage devices for use with Oracle ASM 11.2 on IBM: Linux on System z under SLES*.

Implement UDEV rules when ASM is used.

2.3.3 Oracle Database recommendations

Remember the following points when you are considering the use of an Oracle Database:

- Consider separating redo logs from database files for better performance.
- Make redo logs large enough to reduce frequent log switches.

2.3.4 File system recommendations

Remember the following file system recommendations:

- Disable Logical Volume Manager (LVM) read ahead for database disks.
- In the `initora` file, set `filesystemio_options=`**`'setall'`** to disable Linux cache.

2.3.5 Validation of the I/O subsystem

Use Oracle's Orion tool to test the I/O infrastructure before the database is installed. Orion simulates Oracle Database such as I/O and therefore can induce a load on the I/O subsystem to test its throughput.

To get the latest version of Orion, perform a software-only installation of 11gR2, and the executable file can be found in the `$ORACLE_HOME/bin` directory. Run the **`$ORACLE_HOME/bin/orion -help`** command to review the latest features.

A command to run a test is shown in the following example:

```
./orion_zlinux -run simple -testname mytest -num_disks 2 -duration 30 -simulate raid0
```

For more information, see the *Oracle Database on Linux on System z - Disk I/O Connectivity* white paper, which is available at this website:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102234

2.4 Considerations for network choices

For more information about network choices considerations, see Chapter 3, "Network connectivity options for Oracle on Linux on IBM System z" on page 29.

2.5 Other performance considerations

Experience shows that Oracle on Linux on System z often can run better with less physical memory than on a stand-alone distributed platform.

Consider the following points:

- Does the Oracle Database require all of the memory it had in the non-virtualized environment?
- Optimize use of shared resources by not over-allocating memory and virtual processors to guests.
- Properly prioritize guests by using z/VM share capabilities.
- In a shared environment, look for workloads that do not have to run at the same time. Occasionally, when a number of Oracle Database servers are consolidated, each now virtualized server runs Oracle backups at the same scheduled time, which increases the burden on shared resources. Consider spreading the Oracle backups across the available backup window.

2.6 Considerations for how to run a PoC project

Consider the following points if you are going to run a PoC project to implement an Oracle Database on Linux on System z:

- Engage an IBM System z and Oracle specialist to assist with PoC planning.
- Attend educational opportunities, if possible. IBM provides workshops on this topic many times throughout the year.

  For example, Oracle's E-Business Suite (DB only), PeopleSoft (DB only), and Siebel (DB only) are all certified to run on Linux on System z.
- Obtain IFLs and memory as per the sizing process. Do not acquire zIIPs, zAAPs, or CPs for this environment because only IFLs are necessary.
- Choose an I/O subsystem (that is, ECKD or SCSI) that fits within your environment and performs well because Oracle can generate a significant I/O load.
- Install z/VM and z/VM Performance Toolkit to measure overall usage.
- Install Linux and choose certified levels of SUSE Linux Enterprise Server or Red Hat Enterprise Linux.
- Go to `http://support.oracle.com` (an ID and password are required) and review notes 1306889.1, 1290644.1, and 1308859.1.
- Verify that the required Oracle modules are installed. RPM checkers are available. For more information, see Oracle support note ID 1086769.1.
- Consider the use of VDisk for primary swap spaces.
- Install the latest releases of the Oracle Database - 11gR2 (11.2.0.3) or 10gR2 (10.2.0.5)
- Consider starting with Oracle's ASM versus LVM ext3 files. Use UDEV rules instead of ASMLIB. For more information, see Oracle support note ID 1350008.1 and 1089399.1.
- If you are using ext3, verify that the Oracle Database `init.ora` has the following settings:
  - `filesystemio_options = setall` to enable direct I/O and to bypass Linux cache
  - `disk_asynch_io=true`

- Consider the use of Linux Hugepages for large SGAs (Oracle DB 11gR2 only)

  The more dedicated connections the database has, the larger the page table reduction. However, this choice does feature the limitation that there are no advanced management modules (AMMs).

- Create an appropriate disk multipathing SCSI disk.
- Review the following options to move an Oracle Database to Linux on System z:
  - Use transportable table space or transportable database for metadata when endian formats are the same.
  - Other steps, such as, rman conversions, are required for unlike endian formats.
  - Import/export might be required when the source database is older than 10gR2.

2.6.1 Run PoC testing

Complete the following steps during the PoC testing:

- Collect performance data by enabling the following components:
  - z/VM Performance Toolkit

    You must now think about virtualization versus dedicated resources.

  - **sar** and **iostat** data from theLinux on System z guest (or guests)
  - AWR reports from the Oracle Database
- Review performance reports:
  - z/VM Performance Toolkit
    - Understand CPU, memory, and paging consumption for the LPAR
    - Review virtual machine usage of resources
    - Evaluate I/O performance
  - Verify VDisk usage:
    - Linux that uses **sar** and **iostat** data
    - CPU, memory, swapping, and I/O performance for each guest
  - Oracle AWR report and Oracle Enterprise Manager windows:
    - I/O performance
    - SGA and PGA usage via automatic memory management (see Figure 2-10 on page 22, Figure 2-11 on page 22, and Figure 2-12 on page 23)
    - Normal DBA tuning review
  - Review for poor performing SQL

    Locking and latching

- Rerun PoC if changes are made

Does the PoC validate the initial sizing?

- Does the Oracle Database require all of that memory it had in the non-virtualized environment?
- Should you have a active/passive/stand by setup in the same z/VM?

  Optimize use of resources

- Were the guests properly prioritized regarding other guests in z/VM?

- What workloads are peaking at the same time:
  - CPU peak
  - Memory load
  - I/O subsystem

> **Tip:** Database administrators, Linux administrators, and z/VM system programmers must work as a team in any virtualized environment.

2.7 Successful projects: Is the solution production-ready?

Consider the following points when you are trying to determine of the solution is production-ready:

- Did the PoC validate the initial sizing?

  If not, attempt to resize or use PoC information as the basis for estimation.
- Did the PoC test the following availability requirements that were established during the requirements phase:
  - Stand-alone database
  - RAC with Active/Active or Active/Passive
  - Use of multiple physical System z machines
  - Data Guard for Disaster Recovery
- Is there sufficient IFL capacity, memory, and I/O capacity for production?

  Are you ready to measure capacity usage over the long term?
- Are the latest Oracle patches applied?
- Consider z/VM prioritization to appropriately manage the large number of guests.

3

Network connectivity options for Oracle on Linux on IBM System z

This chapter describes the various options that are available to configure the private interconnect in a shared infrastructure environment for Oracle Real Application Clusters (RAC) with Linux on System z. In addition, the network best practices that are described can be used when application servers are configured to Oracle databases that are running under Linux on System z in a highly available environment.

If a single instance Oracle database without Oracle Clusterware (RAC) is installed, you do not need to configure the private interconnect for RAC. We introduce this topic for planning purposes so you can review your options as you are preparing for your installation.

This chapter focuses primarily on configuring the network on Linux on System z for performance and high availability when you are connecting to another System z system for Oracle RAC and connectivity to Application Servers that are running on machines that are separate than that of the database.

The following High Availability (HA) network connectivity options are described:

- Two Open System Adapter (OSA) network cards that are configured to a z/VM VSWITCH that is defined as active/passive.
- Two OSA cards that are configured to a VSWITCH defined by using z/VM link aggregation (active/active).
- Two OSA cards that are configured with Linux Ethernet bonding.
- Two OSA cards that are configured with Oracle's new Redundant Interconnect feature (Oracle RAC Interconnect only).

The z/VM and Linux configurations for these options are described in detail in 3.6, "Setting up z/VM" on page 38 and section 3.7, "Linux setup for Oracle RAC Interconnect Interfaces" on page 41. The decision to incorporate more HA network solutions is based on the business requirements, hardware availability, and the skill sets available. For example, a site that has strong z/VM skills might decide to use the z/VM VSWITCH options. Sites with stronger DBA skill sets might decide to use the Oracle redundant Interconnect capabilities with Linux on System z.

This chapter includes the following topics:

- Overview
- Network considerations for running Oracle RAC with Linux on System z
- Virtual local area network tagging
- Designing the network configuration for HA with Oracle on Linux on System z
- Oracle RAC recommended configurations for Linux on System z
- Setting up z/VM
- Linux setup for Oracle RAC Interconnect Interfaces
- Notes and observations
- Summary

3.1 Overview

Oracle RAC with Linux on System z was one of the first certified and supported virtualized platforms for running Oracle RAC in a virtual environment with z/VM. Table 3-1 shows the current certification matrix for running Oracle RAC with Linux on System z.

Table 3-1 Supported Virtualization Technologies for Oracle database and RAC product releases[1]

Platform	Virtualization Technology	Operating System	Certified Oracle Single Instance Database Releases	Certified Oracle RAC Databases Releases
IBM System z	z/VM and System z LPARs	Red Hat Enterprise Linux SUSE Linux Enterprise Server	10gR2[a] 11gR2[a]	10gR2[a] 11gR2[a]

a. Oracle RAC is certified to run virtualized with z/VM and by using native logical partitions.

A key component for a successfully running Oracle RAC database is the network interconnect between nodes. The default for Oracle RAC in 11gR2 allows up to 30 seconds of network interruption between nodes before the cluster evicts an unresponsive node.

Performance of the cluster interconnect server connectivity is an important consideration to reduce latencies, lost fragments, and Oracle data blocks between connecting nodes. Table 3-2 shows the certified Oracle RAC interconnect configurations for running RAC on System z.

Table 3-2 RAC Technologies Matrix for Linux Platforms[1]

Platform	Technology categories	Technology	Notes
IBM System z Linux	Server Processor Architecture	IBM System z	Certified and supported on certified distributions of Linux running natively in LPARs or as a guest OS on z/VM virtual machines, deployed on IBM System z 64-bit servers
	Network Interconnect Technologies	VLAN with one System z Ethernet over Gigabit OSA card for two System z HiperSockets	

IBM System z is an ideal platform for consolidating Oracle RAC workloads. One example of this is the unique System Assist Processors (SAP) of System Z. SAP processors are internal System z CPU processors that assist with the offload of network and I/O CPU cycles to the SAP processors from the main processors that an Oracle RAC node might be using.

[1] Source: http://www.oracle.com/technetwork/database/clustering/tech-generic-linux-new-086754.html

CPU offload helps prevent Oracle RAC interconnect CPU starvation wait events. CPU usage still should be monitored to avoid interconnect waits that are related to CPU starvation.

Consolidating databases in a shared System z RAC environment is supported, if the network traffic for the private interconnect is restricted to networks with similar performance and availability characteristics. If the environment requires increased security between RAC clusters, VLAN tagging can be used between distinct Oracle cluster nodes.

3.2 Network considerations for running Oracle RAC with Linux on System z

This section describes some of the architectural requirements for designing an Oracle RAC Interconnect environment on a System z Linux system.

The minimum recommendation for an online transaction processing (OLTP) database is a 1 Gb network. Decision Support/Data Warehouse databases can require greater than a 1 Gb network interface for the Oracle Interconnect. It is important to note that this is the minimum recommendation and monitoring of the network interfaces is needed to ensure that the performance is running at acceptable levels.

For example, 1000 Mbits (1 Gb) is equivalent to 125 MBps. When you exclude header and pause frames, this can equate to a maximum of 118 MBps per 1 Gb network interface. When sharing a network card across multiple databases, the interface must support the peak workload of all the databases. It is recommended that 10 Gb network interfaces are used for shared network workloads.

The network interconnect for Oracle RAC must be on a private network. If you are configuring the private interconnect between System z physical machines, there must be a physical switch configured ideally with one network hop between systems. The private interconnect should also be a private IP address in the range that is shown in the following example:

```
Class A: 10.X.X.X
Class B: 172.(16-31).X.X
Class C: 192.168.X.X     where    (0 <= X <= 255)
```

For example, if the Oracle private interconnect is configured with a public routable IP address, it is possible for other systems to affect the Oracle RAC database's interconnect traffic. Although not mandatory, it is recommended and a best practice to use private IP address ranges for the interconnect configuration.

An Oracle RAC workload sends a mixture of short messages of 256 bytes and database blocks of the database block size for the long messages. Another important consideration is to set the Maximum Transition Unit (MTU) size to be a little larger than the data base block size for the database. To support a larger MTU size, the network infrastructure (switches) should be configured to support Jumbo frames for the Oracle Interconnect network traffic.

Table 3-3 shows the extra number of network reassemblies that are required when the MTU size is not set to a value that is larger than database block size (8K).

Table 3-3 Example of setting the MTU Size to a size that is greater than the DB Block Size

'netstat -s' of interconnect	MTU Size of 1492 (default)	MTU Size of 8992 (with 8K DB block size)
Before reassemblies	43,530,572	1,563,179
After reassemblies	54,281,987	1,565,071
Delta assemblies	**10,751,415**	**1,892**

The smaller MTU results in a higher number of network assemblies. High network reassemblies on the receive and transmit side can result in higher CPU to be used because of the breaking apart and then reassembling of these other network packets.

In the physical switch configuration that is uplinked from the System z machine, it is recommended to prune out the private Oracle Interconnect traffic from the rest of the network.

3.3 Virtual local area network tagging

Larger network environments with many systems are sometimes configured into virtual local area networks (VLANs). VLAN tagging helps address issues such as scalability, security, and network management.

The use of a separate interconnect VLAN for each Oracle RAC cluster helps minimize the impact of network spanning tree events that affect other RAC clusters in the network.

VLAN tagging can be used to segregate the Oracle Cluster interconnect network traffic between Linux guests that are configured for different RAC Clusters. VLAN tagging is fully supported for Oracle Clusterware Interconnect deployments, if the interconnect traffic is segregated.

VLAN tagging is also useful for sharing network interfaces between clusters. Monitoring of shared network interfaces should be done to avoid performance problems with a maximum usage of up to 60% - 70% of the bandwidth of the interfaces.

Other interfaces can be aggregated to help provide more bandwidth to a saturated network interface.

VLAN tagging is configured by setting up the Linux network interfaces files, as described in 3.7, "Linux setup for Oracle RAC Interconnect Interfaces" on page 41. Depending on the type of network switch that is used, configuring of the switch ports might be required.

3.4 Designing the network configuration for HA with Oracle on Linux on System z

Another important consideration when you are designing the network for running Oracle on System z is to review the high availability (HA) requirements for the applications.

The following methods can be used to configure an HA solution for a System z environment that uses a multi-node configuration:

- Virtual Switch (Active/Passive): When one Open System Adapter (OSA) Network port fails, z/VM moves the workload to another OSA card port; z/VM handles the fail over.
- Link Aggregation (Active/Active): Allows up to eight OSA-Express adapters to be aggregated per virtual switch. Each OSA-Express port must be exclusive to the virtual switch (for example, it cannot be shared); z/VM handles the load balancing of the network traffic.
- Linux Bonding: Creates two Linux interfaces (for example, eth1 and eth2) and create a bonded interface bond0 made up of eth1 and eth2, which the application uses. Linux can be configured in various ways to handle various failover scenarios.
- Oracle HAIP: New in 11.2.0.2+, Oracle can have up to four Private Interconnect interfaces to load balance Oracle RAC Interconnect traffic. Oracle handles the load balancing and is exclusive to Oracle Clusterware implementations.

Figure 3-1 shows a shared Active/Passive VSWITCH configuration. An Active/Passive VSWITCH configuration provides HA if an OSA card fails. The failover time to the other redundant OSA card must be considered when you are working in a Oracle RAC configuration to not affect the performance of the cluster.

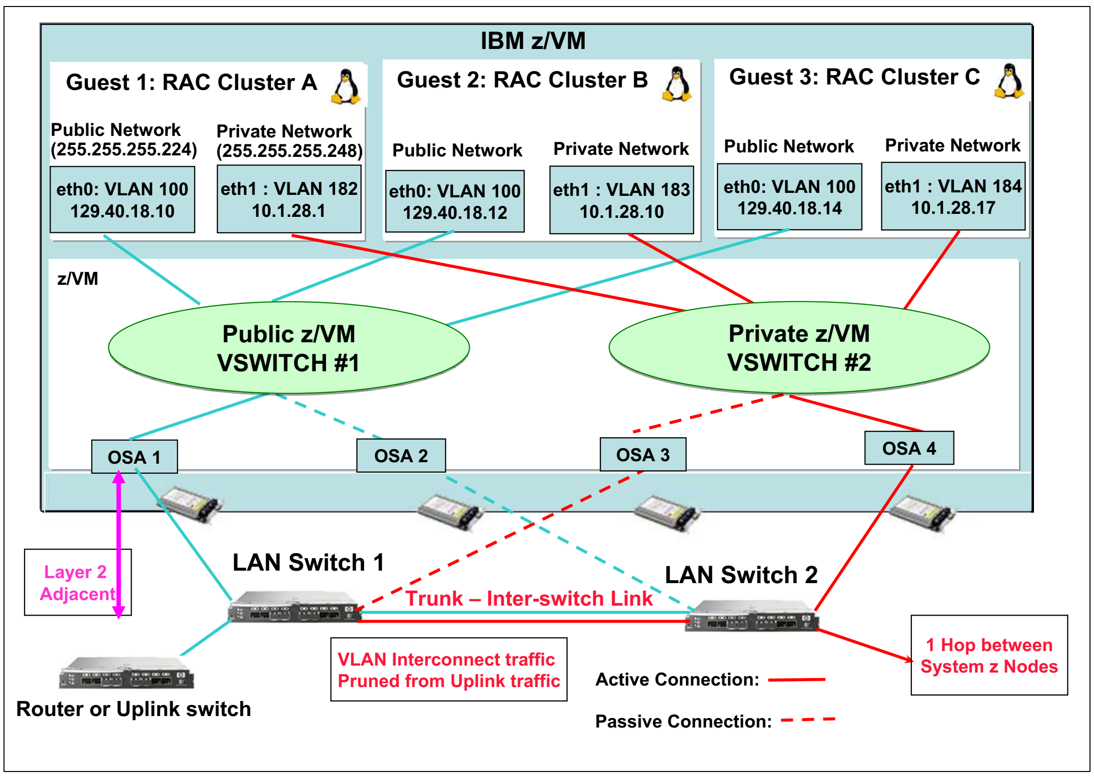


Figure 3-1 Shared Private Virtual Switch across Multiple System z systems (CPCs)

Similar to the Active/Passive VSWITCH configuration is z/VM Link Aggregation. z/VM Link Aggregation allows for up to eight OSA cards to be aggregated to provide failover and extra bandwidth capabilities. One restriction with link aggregation is that any OSA ports that are defined with Link Aggregation must be exclusive to the VSWITCH in the LPAR and cannot be shared with other LPARs or VSWITCHES in that LPAR.

Figure 3-2 shows a VSWITCH active/active Link Aggregation configuration. An entire OSA-Express card is placed in a state of exclusivity by the virtual switch for Link Aggregation operations. The VSWITCH attempts to load balance network traffic across all the available ports automatically in the Link Aggregation port group.

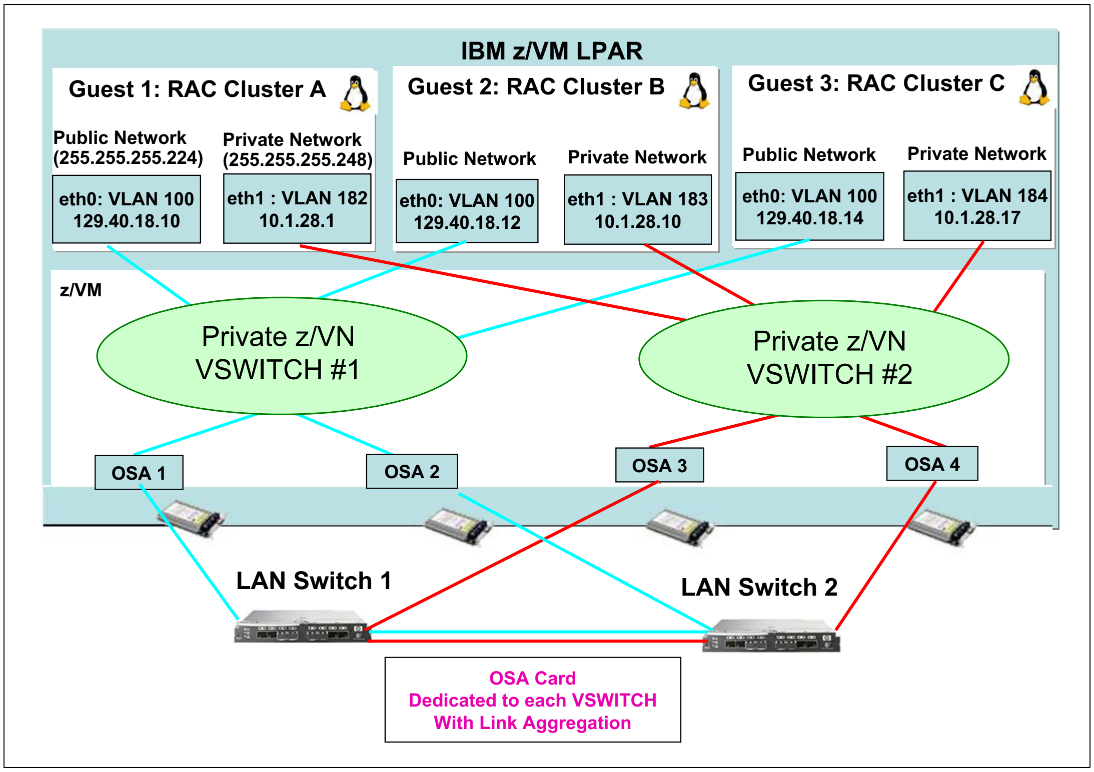


Figure 3-2 Active/Active z/VM Virtual Switch Link Aggregation configuration

Another HA option is to use Linux bonding across multiple network OSA cards, as shown in Figure 3-3 on page 36. If the network configuration fails, the other network interface that uses a separate network/OSA card and switch provides the failover capability for the Oracle Interconnect.

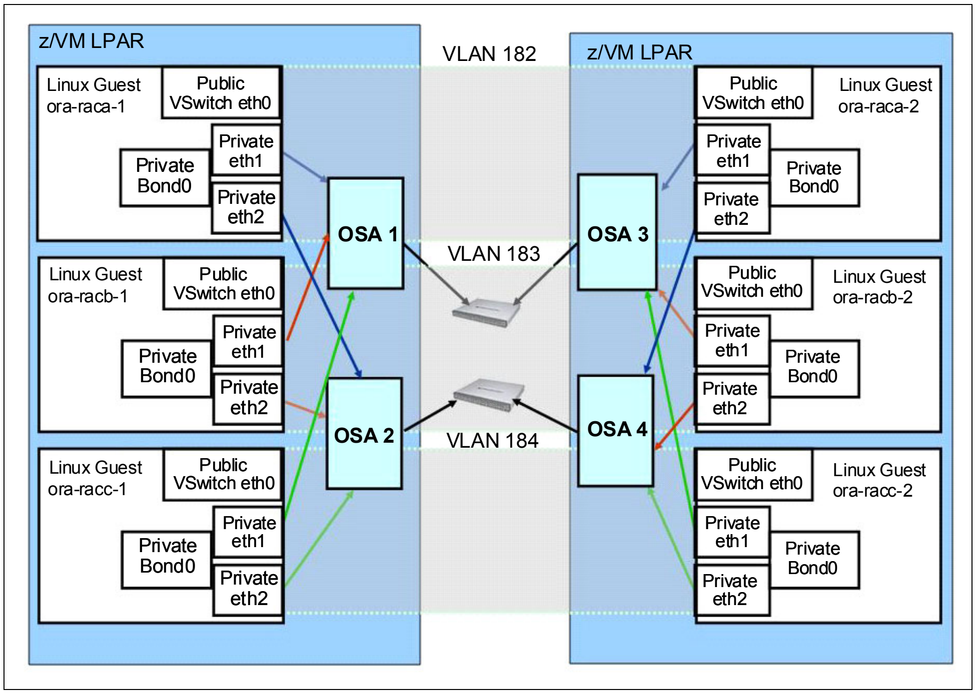


Figure 3-3 Linux Bonding Oracle Interconnect across Multiple System z systems (CPCs)

The final HA option to consider is Oracle's new 11gR2 HAIP capabilities. Similar to Linux bonding, two separate Linux OSA channels that are configured on separate cards and switches are presented to the Linux guest (for example, as eth1 and eth2). Oracle HAIP provides the failover and load balances the Oracle Interconnect traffic across both interconnect connections (eth1 and eth2), as shown on Figure 3-4.

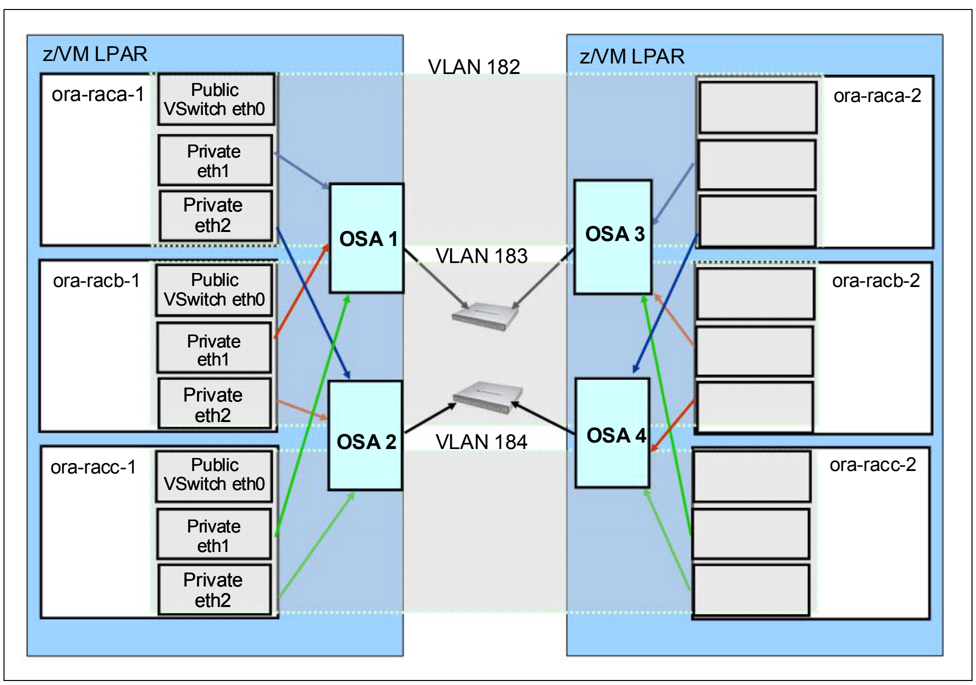


Figure 3-4 Oracle HAIP Interconnect configuration

3.5 Oracle RAC recommended configurations for Linux on System z

Oracle RAC nodes can be configured in many ways in a System z environment. System z's mainframe heritage of high redundancy and availability allow for greater flexibility in these design considerations.

In Chapter 2, "Getting started on a proof of concept project for Oracle Database on Linux on System z" on page 13, various network architectures were introduced that were based on availability.

Table 3-4 provides the recommended network configurations that are based on the architectures available for System z.

Table 3-4 Recommended Oracle RAC network configurations for Linux on System z

Architecture	Oracle Private Network (interconnect)	Oracle Public Network
All z/VM Linux guests in one LPAR	► Private Layer2 VSwitch Guest LAN OSA recommended ► Real layer 2 HiperSocket possible ► Guest LAN HiperSocket not supported	► Shared Public VSwitch recommended ► Shared or dedicated OSA card is possible
z/VM Linux guests on different LPARs	► Real Layer 2 HiperSocket recommended ► Private Layer 2 Gigabit OSA card possible	► Shared Public VSwitch recommended ► Shared or dedicated OSA card
z/VM Linux guests on different physical machines	Private Layer 2 Gigabit OSA card recommended with physical switch in between (one hop)	Dedicated OSA card Possible

If business requirements allow, HiperSocket across multiple LPARs on the same System z machine (CPC) is recommended for production environments because this provides protection from any LPAR or z/VM maintenance and key performance benefits with HiperSockets. HiperSockets can provide low latencies in the range of 50 microseconds for a 250 byte packet compared to 300 microseconds for an OSA network card.

3.5.1 Other considerations: Using Oracle Server Pools

Another method of sharing a network interface between systems is to use an Oracle feature called Server Pools.

Server Pools create a logical division of the cluster into pools of servers with many Linux Guests using the same cluster interconnect, as shown in Figure 3-5 on page 38.

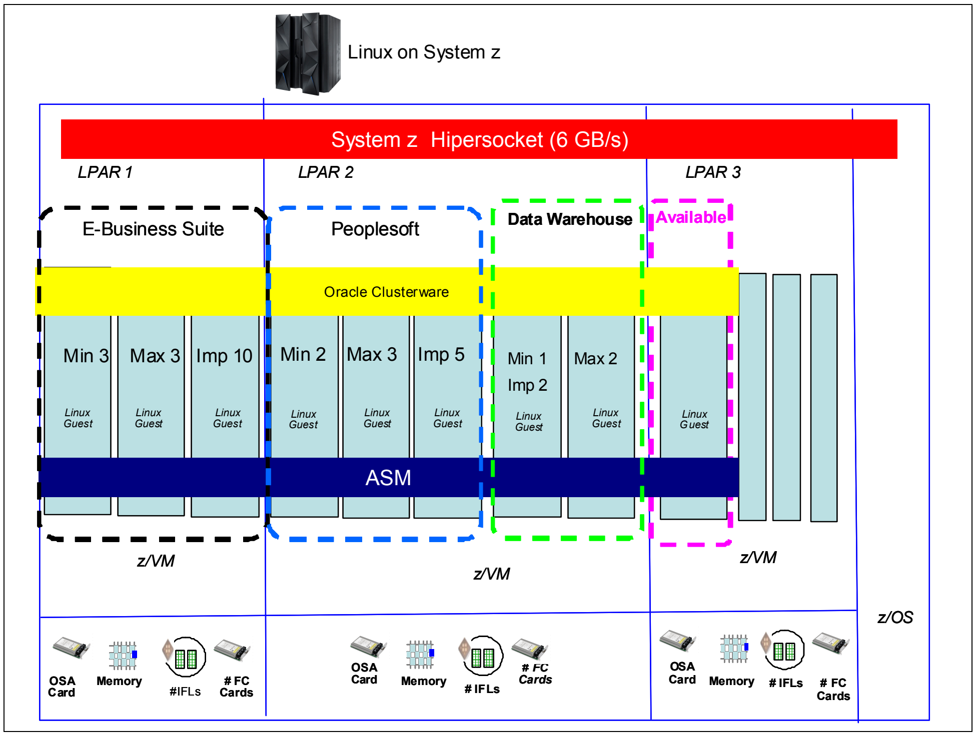


Figure 3-5 Using Server Pools

Cluster management that uses Oracle Server Pools can be used for planned or unexpected outages. For example, if one of the systems that is shown in Figure 3-5 needs maintenance in LPAR 1, Oracle Clusterware migrates the instance to a free available node in LPAR 3. The relocation occurs while the database is fully operational to users.

Oracle Server Pools work well in a System z environment because resources such as network interfaces, IFL (CPU) capacity, and disk storage are shared. When the database moves to another node, the resources can be used by the other nodes.

3.6 Setting up z/VM

The following section describes the z/VM setup steps when you are configuring a private VSWITCH with VLAN tagging for Oracle RAC systems.

To make use of VLAN tagging and failover, the following commands are used to dynamically define and grant access to a separately created Layer 2 VSWITCH:

```
DEFINE VSWITCH ORACHPR ETHERNET RDEV 1100 3870  VLAN 100 PORTTYPE TRUNK
DEFINE NIC 1420 QDIO DEV 3
SET VSWITCH ORACHPR GRANT MAINT
COUPLE 1420 TO SYSTEM ORACHPR
SET VSWITCH ORACHPR GRANT ORARACA2 VLAN 182
SET VSWITCH ORACHPR GRANT ORARACB2 VLAN 183
SET VSWITCH ORACHPR GRANT ORARACC2 VLAN 184
```

To change the MTU size for the interconnect interfaces to a value that is slightly larger than the 8K database block size, we modified the VSWITCH MTU size to 8992 from the default 1492 size, as shown in the following example:

```
SET VSWITCH ORACHPR PATHMTUD VAL 8992
```

To make the change permanent, the SYSTEM CONFIG file that is shown in Example 3-1 requires updating.

Example 3-1 Sample SYSTEM CONFIG for VLAN Tagging

```
/*-------------------------------------------------------------------------------
/*                  Perf  VSWITCH Definition
/*-------------------------------------------------------------------------------
DEFINE VSWITCH ORACHPR ETHERNET VLAN 100 RDEV 1100 3870 PORTTYPE TRUNK
MODIFY VSWITCH ORACHPR GRANT MAINT
MODIFY VSWITCH ORACHPR GRANT ORARACA2 VLAN 182
MODIFY VSWITCH ORACHPR GRANT ORARACB2 VLAN 183
MODIFY VSWITCH ORACHPR GRANT ORARACC2 VLAN 184
VMLAN MACPREFIX 021112 USERPREFIX 020000
```

For Ethernet vswitches, a unique MACPREFIX must be defined for each switch, even if they exist on different CPCs if they must communicate with each other. Example 3-2 shows the access list of a vswitch with VLAN tagging.

Example 3-2 VLAN Tagging access list

```
q vswitch orachpr acc
VSWITCH SYSTEM ORACHPR  Type: QDIO    Connected: 4    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    ETHERNET                  Accounting: OFF
  USERBASED
  VLAN Aware  Default VLAN: 0100    Default Porttype: Trunk   GVRP: Enabled
              Native  VLAN: 0001    VLAN Counters: OFF
  MAC address: 02-11-12-00-00-1E    MAC Protection: Uspecified
  IPTimeout: 5         QueueStorage: 8
  Isolation Status: OFF
    Authorized userids:
      MAINT    Porttype: Trunk  VLAN: 0100
      ORARACA2 Porttype: Trunk  VLAN: 0182
      ORARACB2 Porttype: Trunk  VLAN: 0183
      ORARACC2 Porttype: Trunk  VLAN: 0184
 Uplink Port:
  State: Ready
  PMTUD setting: EXTERNAL   PMTUD value: 8992
  RDEV: 1100.P00 VDEV: 0600 Controller: DTCVSW1
  RDEV: 3870.P00 VDEV: 0600 Controller: DTCVSW2  BACKUP
Ready; T=0.01/0.01 14:10:43
```

Example 3-3 shows a detailed configuration for the Private Interconnect VSWITCH with VLAN tagging.

Example 3-3 Detailed VLAN Tagging access list

```
q vswitch orachpr det
VSWITCH SYSTEM ORACHPR  Type: QDIO    Connected: 4    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    ETHERNET                  Accounting: OFF
  USERBASED
  VLAN Aware  Default VLAN: 0100    Default Porttype: Trunk   GVRP: Enabled
              Native  VLAN: 0001    VLAN Counters: OFF
  MAC address: 02-11-12-00-00-1E    MAC Protection: Uspecified
  IPTimeout: 5        QueueStorage: 8
  Isolation Status: OFF
Uplink Port:
  State: Ready
  PMTUD setting: EXTERNAL   PMTUD value: 8992
  RDEV: 1100.P00 VDEV: 0600 Controller: DTCVSW1
    Uplink Port Connection:
      RX packets: 39638266   Discarded: 515       Errors: 0
      TX packets: 42728123   Discarded: 0         Errors: 0
      RX Bytes: 28111347573          TX Bytes: 32500322442
      Device: 0600  Unit: 000   Role: DATA       Port: 2049
        Unicast IP Addresses:
          169.254.88.208       MAC: 02-11-13-00-00-0C Remote
          169.254.126.130      MAC: 02-11-13-00-00-09 Remote
          169.254.204.100      MAC: 02-11-13-00-0A-28 Remote
  RDEV: 3870.P00 VDEV: 0600 Controller: DTCVSW2  BACKUP
 Adapter Connections:                            Connected: 4
    Adapter Owner: MAINT    NIC: 1420.P00 Name: UNASSIGNED  Type: QDIO
      Porttype: Trunk
      Port: 0001
    Adapter Owner: ORARACA2 NIC: 1420.P00 Name: UNASSIGNED  Type: QDIO
      Porttype: Trunk
      RX packets: 20220767   Discarded: 515       Errors: 0
      TX packets: 15132101   Discarded: 0         Errors: 34
      RX Bytes: 15524758705          TX Bytes: 8166344863
      Device: 1422  Unit: 002   Role: DATA       Port: 0010
      VLAN: 0182
      Options: Ethernet Broadcast
        Unicast MAC Addresses:
          02-11-12-00-00-38 IP: 169.254.28.198
        Multicast MAC Addresses:
          01-00-5E-00-00-01
          01-00-5E-00-00-FB
          01-00-5E-00-01-00
```

When you are switching from this setup to use Link Aggregation, it is important to remember that the entire OSA card is used for a port group, not just a subset of three addresses.

If you try to use a range of addresses on this card for another function, then you see “exclusive use errors.”

The network administrator also must change the physical switch to enable port groups. Otherwise, you see a message stating: “LACP (Link Aggregation Control Protocol) was not enabled on partner.”

The advantages of a Link Aggregation setup include increased throughput, resiliency, and bandwidth because I/O that is sent from one of the interfaces (in this case, 1100 interface) can be returned via the other interface, which in our case was 3870.

3.7 Linux setup for Oracle RAC Interconnect Interfaces

There were four different private interconnect scenarios tested. Scenarios 1 and 2 that were based on VSWITCH are not apparent to Linux on System z. The Linux guests recognize the VSWITCHES as the same device address whether it was configured active/passive or with link aggregation. The following scenarios were tested:

- Two OSA cards configured to a VSWITCH defined as active/passive.
- Two OSA cards configured to a VSWITCH defined by using link aggregation.
- Two physical OSA cards that Linux on System z used to perform Ethernet Bonding.
- Two physical OSA cards given to Oracle to implement redundant interconnect HAIP feature.

The following steps were used to define the various VLAN tagged network interfaces (VSWITCH, Ethernet Bonding, and HAIP). After the z/VM definitions are in place, we assume that the following points:

- The virtual NIC device address of `0.0.1410-0.0.1412` was used for our public interface and was always named `eth0`.
- The private z/VM VSWITCH device address `0.0.1420-0.0.1422` was used for the Oracle Interconnect and was always called `eth1`.
- The dedicated OSA card device addresses varied between clusters and physical CPCs. There were always two physical OSA cards defined to each of our Linux guests and they were named `eth2` and `eth3`. Table 3-5 shows how these interfaces were used in our study.

Table 3-5 Consolidated table of all Linux Interfaces Used

Usage	Interface Name	Description
Public Interface	eth0	z/VM VSWITCH (separate from interconnect)
Private interconnect	eth1	z/VM VSWITCH
Private interconnect	eth2	Ethernet bonding / HAIP
Private interconnect	eth3	Ethernet bonding / HAIP

Table 3-5 also shows the multiple Linux interfaces that were configured. If you are using Linux bonding `eth0` and `eth1` can be bonded for the Public Interface. We configured two Private Interfaces (`eth1` with VSWITCH Aggregation) and used `eth2` and `eth3` for the Bonded private interconnect solutions.

For the private interconnect, we used two physical OSA cards on each CPC. To segregate different networks (per RAC cluster) that use the same NIC (such as a VSWITCH), we configured VLAN tagging.

This process required the following tasks:

- The system router that the OSA cards of the CPC were connected to needed to be trunked into separate subnets. Each subnet was given a VLAN tag. The VLAN tag identifiers that were used in this study were `182`, `183` and `184`. Each tag was for a different Oracle RAC cluster.
- The z/VM VSWITCH had to be defined as VLAN aware.
- The z/VM VSWITCH must be defined as Layer 2 for an Oracle Interconnect to function properly.
- Linux configuration files also set the physical OSA cards used for Ethernet bonding and HAIP as Layer 2.
- The MTU size had to match throughout the environment, from system router to z/VM to Linux configuration.

3.7.1 Setting up the private VSWITCH

It is assumed that the VSWITCH device address is online to the Linux system. We use device `0.0.1420-0.0.1422` as an example.

SUSE Linux Enterprise Server 11

Complete the following steps to set up a VSWITCH in SUSE Linux Enterprise Server 11:

1. Add the VSWITCH in `/etc/udev/rules.d`
2. Ensure that Layer2 is enabled for the Oracle RAC interconnect. Configure the VSWITCH by using the `ifcfg` scripts. A standard and VLAN tagged script is shown. The `ifcfg` scripts can be found under `/etc/sysconfig/network/` directory.
3. Reboot the system and issue the `ifconfig` command.
4. Add the VSWITCH in `/etc/udev/rules.d/`. Each network device is placed under the following directory:

```
-rw-r--r-- 1 root root 1661 Jul 17 10:52 51-qeth-0.0.1410.rules
-rw-r--r-- 1 root root 1661 Sep 14 01:19 51-qeth-0.0.1420.rules
-rw-r--r-- 1 root root 1661 Sep 12 09:14 51-qeth-0.0.1423.rules
```

The value of ATTR{layer2}="1"is required for the Oracle Interconnect, as shown when running the following command (note the second last line):

```
$ cat 51-qeth-0.0.1420.rules
#Configure qeth device at 0.0.1420/0.0.1421/0.0.1422
ACTION=="add", SUBSYSTEM=="drivers", KERNEL=="qeth", IMPORT{program}="collect
0.0.1420 %k 0.0.1420 0.0.1421 0.0.1422 qeth"
ACTION=="add", SUBSYSTEM=="ccw", KERNEL=="0.0.1420", IMPORT{program}="collect
0.0.1420 %k 0.0.1420 0.0.1421 0.0.1422 qeth"
ACTION=="add", SUBSYSTEM=="ccw", KERNEL=="0.0.1421", IMPORT{program}="collect
0.0.1420 %k 0.0.1420 0.0.1421 0.0.1422 qeth"
ACTION=="add", SUBSYSTEM=="ccw", KERNEL=="0.0.1422", IMPORT{program}="collect
0.0.1420 %k 0.0.1420 0.0.1421 0.0.1422 qeth"
ACTION=="remove", SUBSYSTEM=="drivers", KERNEL=="qeth",
IMPORT{program}="collect --remove 0.0.1420 %k 0.0.1420 0.0.1421 0.0.1422 qeth"
ACTION=="remove", SUBSYSTEM=="ccw", KERNEL=="0.0.1420",
IMPORT{program}="collect --remove 0.0.1420 %k 0.0.1420 0.0.1421 0.0.1422 qeth"
ACTION=="remove", SUBSYSTEM=="ccw", KERNEL=="0.0.1421",
IMPORT{program}="collect --remove 0.0.1420 %k 0.0.1420 0.0.1421 0.0.1422 qeth"
```

```
ACTION=="remove", SUBSYSTEM=="ccw", KERNEL=="0.0.1422",
IMPORT{program}="collect --remove 0.0.1420 %k 0.0.1420 0.0.1421 0.0.1422 qeth"
TEST=="[ccwgroup/0.0.1420]", GOTO="qeth-0.0.1420-end"
ACTION=="add", SUBSYSTEM=="ccw", ENV{COLLECT_0.0.1420}=="0",
ATTR{[drivers/ccwgroup:qeth]group}="0.0.1420,0.0.1421,0.0.1422"
ACTION=="add", SUBSYSTEM=="drivers", KERNEL=="qeth",
ENV{COLLECT_0.0.1420}=="0",
ATTR{[drivers/ccwgroup:qeth]group}="0.0.1420,0.0.1421,0.0.1422"
LABEL="qeth-0.0.1420-end"
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.1420", ATTR{portno}="0"
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.1420", ATTR{layer2}="1"
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.1420", ATTR{online}="1"
```

5. Configure the VSWITCH by using the **ifcfg** scripts. The following example is a standard **ifcfg** script (settings such as netmask and broadcast differ based on your network). An MTU size of 8992 was used throughout the network (VSWITCH, OSA cards and network switch):

```
ora-raca-1:/etc/sysconfig/network # cat ifcfg-eth1
BOOTPROTO='static'
IPADDR='10.1.28.1/29'
NETMASK='255.255.255.248'
BROADCAST='10.1.28.7'
STARTMODE='onboot'
MTU='8992'
NAME='OSA Express Network card (0.0.1420)'
```

6. Configure the VSWITCH by using the **ifcfg** scripts for VLAN tagging. The following examples are the two `ifcfg` scripts that were used. All IP information was removed from `ifcfg-eth1` and placed into `ifcfg-eth1_182` (182 is our VLAN tag). The VLAN tag is given to you by your network administrator. Separate IP information can be placed into the `ifcfg-eth1` script for non-VLAN tagging use. No additions are needed under `/etc/udev/rules.d`:

```
ora-raca-1:/etc/sysconfig/network # cat ifcfg-eth1
BOOTPROTO='static'
STARTMODE='onboot'
MTU='8992'
NAME='OSA Express Network card (0.0.1420)'

ora-raca-1:/etc/sysconfig/network # cat ifcfg-eth1_182
BOOTPROTO='static'
ETHERDEVICE='eth1'
IPADDR='10.1.28.1/29'
NETMASK='255.255.255.248'
BROADCAST='10.1.28.7'
STARTMODE='auto'
MTU='8992'
NAME='OSA Express Network card (0.0.1420)'
```

> **Important:** The `ETHERDEVICE` statement is required on SUSE Linux Enterprise Server for VLAN tagging and is the link back to the `eth1` device.

7. Reboot the system and run **ifconfig** to verify (only the `eth1` interfaces are shown here):

```
ora-raca-1:~ # ifconfig
eth1      Link encap:Ethernet  HWaddr 02:00:00:6F:77:CF
          inet6 addr: fe80::ff:fe6f:77cf/64 Scope:Link
```

```
          UP BROADCAST RUNNING MULTICAST  MTU:8992  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:1254 (1.2 Kb)

eth1_182  Link encap:Ethernet  HWaddr 02:00:00:6F:77:CF
          inet addr:10.1.28.1  Bcast:10.1.28.7  Mask:255.255.255.248
          inet6 addr: fe80::ff:fe6f:77cf/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:8992  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:578 (578.0 b)
```

Red Hat Enterprise Linux 6

Complete the following steps to setup a VSWITCH in Red Hat Enterprise Linux 6 (RHEL6):

1. Configure the VSWITCH and ensure Layer 2 in the `ifcfg` scripts in the `/etc/sysconfig/network-scripts/` directory. A standard and VLAN tagged script is shown in steps 3 and 4 of this procedure.
2. Reboot the system and run the **`ifconfig`** command to verify.
3. Configure the VSWITCH by using the `ifcfg` scripts. The following example is a standard `ifcfg` script (settings such as netmask and broadcast differ based on your network). An MTU size of 8992 was used throughout the network (VSWITCH, OSA cards and network switch):

```
[root@ora-racb-1 network-scripts]# cat ifcfg-eth1
DEVICE="eth1"
BOOTPROTO="static"
IPADDR="10.1.28.9"
MTU="8992"
NETMASK="255.255.255.248"
NETTYPE="qeth"
NM_CONTROLLED="yes"
ONBOOT="yes"
OPTIONS="layer2=1"
SUBCHANNELS="0.0.1420,0.0.1421,0.0.1422"
```

4. Configure the VSWITCH by using the `ifcfg` scripts for VLAN tagging. The following examples are the two `ifcfg` scripts used. All IP information was removed from `ifcfg-eth1` and placed into `ifcfg-eth1_183` (183 is our VLAN tag for our RHEL6 RAC cluster). The VLAN tag is given to you by your network administrator. Separate IP information can be placed into the `ifcfg-eth1` script for non-VLAN tagging use:

```
[root@ora-racb-1 network-scripts]# cat ifcfg-eth1
DEVICE="eth1"
BOOTPROTO="static"
NETTYPE="qeth"
NM_CONTROLLED="yes"
ONBOOT="yes"
OPTIONS="layer2=1"
MTU="8992"
SUBCHANNELS="0.0.1420,0.0.1421,0.0.1422"
```

```
[root@ora-racb-1 network-scripts]# cat ifcfg-eth1_183
DEVICE="eth1_183"
BOOTPROTO="static"
IPADDR="10.1.28.9"
MTU="8992"
NETMASK="255.255.255.248"
VLAN="yes"
```

> **Important:** The `VLAN=yes` statement is required on RHEL6 for VLAN tagging. Also, the `DEVICE` statement is the link back to the `eth1` device. The VLAN tag is appended to the name.

5. Reboot the system and run the **`ifconfig`** command to verify (only the `eth1` interfaces are shown here):

```
[root@ora-racb-1 network-scripts]# ifconfig
eth1      Link encap:Ethernet  HWaddr 02:11:13:00:13:85
          inet6 addr: fe80::11:13ff:fe00:1385/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:8992  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:4867 (4.7 KiB)

eth1_183  Link encap:Ethernet  HWaddr 02:11:13:00:13:85
          inet addr:10.1.28.9  Bcast:10.1.28.15  Mask:255.255.255.248
          inet6 addr: fe80::11:13ff:fe00:1385/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:8992  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:28 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:4289 (4.1 KiB)
```

3.7.2 Setting up Ethernet bonding

Ethernet bonding is an effective way to increase bandwidth and reliability. In our example, two separate OSA cards were bonded. For our purposes, the device addresses for the two OSA cards are `0.0.1303-0.0.1305` and `0.0.1423-0.0.1425`

SUSE Linux Enterprise Server 11

Complete the following steps to setup Ethernet bonding in SUSE Linux Enterprise Server 11:

1. Add the two OSA card devices to the `/etc/udev/rules.d` directory.
2. Ensure that Layer2 is enabled for the Oracle RAC Interconnect.
3. Configure each OSA card by using the `ifcfg` scripts. The VLAN tag example is used. The `ifcfg` scripts can be found in the `/etc/sysconfig/network/` directory.
4. Reboot the system and issue the **`ifconfig`** command.
5. Add the following OSA device addresses to the `/etc/udev/rules.d/` directory (each network device is placed under this directory):

```
ora-raca-1:/etc/udev/rules.d # ls -la 51-qeth*
-rw-r--r-- 1 root root 1661 Jul 30 16:12 51-qeth-0.0.1303.rules
-rw-r--r-- 1 root root 1661 Jul 17 10:52 51-qeth-0.0.1410.rules
-rw-r--r-- 1 root root 1661 Sep 14 01:19 51-qeth-0.0.1420.rules
```

```
-rw-r--r-- 1 root root 1661 Sep 12 09:14 51-qeth-0.0.1423.rules
```

The contents of the file are no different than that of a VSWITCH other than the device addresses. Ensure that `layer2` parameter is set to 1. The contents for the 1303 OSA device is shown in the following example (note the second to last line):

```
ora-raca-1:/etc/udev/rules.d # cat 51-qeth-0.0.1303.rules
# Configure qeth device at 0.0.1303/0.0.1304/0.0.1305
ACTION=="add", SUBSYSTEM=="drivers", KERNEL=="qeth", IMPORT{program}="collect
0.0.1303 %k 0.0.1303 0.0.1304 0.0.1305 qeth"
ACTION=="add", SUBSYSTEM=="ccw", KERNEL=="0.0.1303", IMPORT{program}="collect
0.0.1303 %k 0.0.1303 0.0.1304 0.0.1305 qeth"
ACTION=="add", SUBSYSTEM=="ccw", KERNEL=="0.0.1304", IMPORT{program}="collect
0.0.1303 %k 0.0.1303 0.0.1304 0.0.1305 qeth"
ACTION=="add", SUBSYSTEM=="ccw", KERNEL=="0.0.1305", IMPORT{program}="collect
0.0.1303 %k 0.0.1303 0.0.1304 0.0.1305 qeth"
ACTION=="remove", SUBSYSTEM=="drivers", KERNEL=="qeth",
IMPORT{program}="collect --remove 0.0.1303 %k 0.0.1303 0.0.1304 0.0.1305 qeth"
ACTION=="remove", SUBSYSTEM=="ccw", KERNEL=="0.0.1303",
IMPORT{program}="collect --remove 0.0.1303 %k 0.0.1303 0.0.1304 0.0.1305 qeth"
ACTION=="remove", SUBSYSTEM=="ccw", KERNEL=="0.0.1304",
IMPORT{program}="collect --remove 0.0.1303 %k 0.0.1303 0.0.1304 0.0.1305 qeth"
ACTION=="remove", SUBSYSTEM=="ccw", KERNEL=="0.0.1305",
IMPORT{program}="collect --remove 0.0.1303 %k 0.0.1303 0.0.1304 0.0.1305 qeth"
TEST=="[ccwgroup/0.0.1303]", GOTO="qeth-0.0.1303-end"
ACTION=="add", SUBSYSTEM=="ccw", ENV{COLLECT_0.0.1303}=="0",
ATTR{[drivers/ccwgroup:qeth]group}="0.0.1303,0.0.1304,0.0.1305"
ACTION=="add", SUBSYSTEM=="drivers", KERNEL=="qeth",
ENV{COLLECT_0.0.1303}=="0",
ATTR{[drivers/ccwgroup:qeth]group}="0.0.1303,0.0.1304,0.0.1305"
LABEL="qeth-0.0.1303-end"
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.1303", ATTR{portno}="0"
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.1303", ATTR{layer2}="1"
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.1303", ATTR{online}="1"
```

6. Configure both OSA devices by using the `ifcfg` scripts (the following examples are `ifcfg` scripts for two OSA qeth devices that are bonded to a third interfaced named `bond0`. The interface names `eth2` and `eth3` were used. All IP information was removed from the `ifcfg-eth2`, `ifcfg-eth3` and `ifcfg-bond0` files. VLAN tagging is added and a `bond0_182` is created. Separate IP information can be placed into the `ifcfg-bond0` script for non-VLAN tagging use. An MTU size of 8992 was used throughout the network (VSWITCH, OSA cards and network switch):

```
ora-raca-1:/etc/sysconfig/network # cat ifcfg-eth2
BOOTPROTO='static'
STARTMODE='onboot'
ETHTOOL_OPTIONS=''
INTERFACETYPE='qeth'
USERCONTROL='no'
MTU='8992'
NAME='OSA Express Network card (0.0.1303)'

ora-raca-1:/etc/sysconfig/network # cat ifcfg-eth3
BOOTPROTO='static'
STARTMODE='onboot'
ETHTOOL_OPTIONS=''
INTERFACETYPE='qeth'
USERCONTROL='no'
```

```
MTU='8992'
NAME='OSA Express Network card (0.0.1423)'

ora-raca-1:/etc/sysconfig/network # cat ifcfg-bond0
BONDING_MASTER='yes'
BONDING_MODULE_OPTS='mode=0 miimon=100'
BONDING_SLAVE0='eth2'
BONDING_SLAVE1='eth3'
BOOTPROTO='static'
STARTMODE='auto'
MTU='8992'
NAME='OSA Express Network card (0.0.1423 and 1303 bond)

ora-raca-1:/etc/sysconfig/network # cat ifcfg-bond0_182
BOOTPROTO='static'
ETHERDEVICE='bond0'
IPADDR='10.1.28.3/29'
NETMASK='255.255.255.248'
BROADCAST='10.1.28.7'
STARTMODE='auto'
MTU='8992'
NAME='OSA Express Network card (0.0.1423 and 1303 bond)'
```

Important: The `ETHERDEVICE` statement is required on SUSE Linux Enterprise Server for VLAN tagging and is the link back to the `bond0` device.

For bonding options, the mode is the bonding policy such as round robin, active/backup, and so on. The default is 0 (round robin). The `miimon` option specifies how often each slave is monitored for link failures.

Also, in the `ifcfg-bond0` file the `BONDING_SLAVE0` and `BONDING_SLAVE1` statements link the bond back to the `eth2` and `eth3` devices.

7. Reboot the system and run the **`ifconfig`** command to verify (only the `eth2`, `eth3`, `bond0`, and `bond0_182` interfaces are shown here):

```
ora-raca-1:~ # ifconfig
bond0     Link encap:Ethernet  HWaddr 02:00:00:8F:36:87
          inet6 addr: fe80::ff:fe8f:3687/64 Scope:Link
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:8992  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:19 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:136 (136.0 b)  TX bytes:1582 (1.5 Kb)

bond0_182 Link encap:Ethernet  HWaddr 02:00:00:8F:36:87
          inet addr:10.1.28.3  Bcast:10.1.28.7  Mask:255.255.255.248
          inet6 addr: fe80::ff:fe8f:3687/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:8992  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:128 (128.0 b)  TX bytes:438 (438.0 b)
```

```
eth2      Link encap:Ethernet  HWaddr 02:00:00:8F:36:87
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:8992  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:136 (136.0 b)  TX bytes:844 (844.0 b)

eth3      Link encap:Ethernet  HWaddr 02:00:00:8F:36:87
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:8992  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:738 (738.0 b)
```

Red Hat Enterprise Linux 6

Complete the following steps to setup Ethernet bonding in RHEL 6:

1. Configure two OSA card devices and ensure Layer 2 is in the `ifcfg` scripts in the `/etc/sysconfig/network-scripts/` directory. The VLAN tag example is used.
2. Reboot the system and run the **`ifconfig`** command
3. Configure the OSA devices by using the `ifcfg` scripts. The following examples are `ifcfg` scripts for two OSA qeth devices that are bonded to a third interfaced named `bond0`. The interface names `eth2` and `eth3` were used. All IP information was removed from the `ifcfg-eth2`, `ifcfg-eth3`, and `ifcfg-bond0` files. VLAN tagging is added and a `bond0_183` is created. Separate IP information can be placed into the `ifcfg-bond0` script for non-VLAN tagging use. The interface names of `eth2` and `eth3` were used. An MTU size of 8992 was used throughout the network (VSWITCH, OSA cards and network switch):

```
[root@ora-racb-1 network-scripts]# cat ifcfg-eth2
DEVICE="eth2"
BOOTPROTO="static"
NETTYPE="qeth"
NM_CONTROLLED="no"
MASTER="bond0"
SLAVE="yes"
ONBOOT="yes"
OPTIONS="layer2=1"
MTU="8992"
SUBCHANNELS="0.0.1429,0.0.142a,0.0.142b"

[root@ora-racb-1 network-scripts]# cat ifcfg-eth3
DEVICE=eth3
BOOTPROTO="static"
NETTYPE="qeth"
NM_CONTROLLED="no"
MASTER="bond0"
SLAVE="yes"
ONBOOT="yes"
OPTIONS="layer2=1"
MTU="8992"
SUBCHANNELS="0.0.1309,0.0.130a,0.0.130b"

[root@ora-racb-1 network-scripts]# cat ifcfg-bond0
DEVICE=bond0
ONBOOT=yes
```

```
BOOTPROTO=none
USERCTL=no
MTU=8992
BONDING_OPTS='mode=0 miimon=100'

[root@ora-racb-1 network-scripts]# cat ifcfg-bond0_183
DEVICE=bond0_183
BOOTPROTO=static
IPADDR=10.1.28.11
MTU="8992"
NETMASK=255.255.255.248
VLAN=yes
```

> **Tips:** The `VLAN=yes` statement is required on RHEL6 VLAN tagging.
>
> Different from SUSE Linux Enterprise Server, each `ifcfg` script that is used for bonding states whether that device is a slave and who the master is (in our case, `bond0`).
>
> For bonding options, the mode is the bonding policy, such as round robin, active/backup, and so on. The default is 0 (round robin). The `miimon` option specifies how often each slave is monitored for link failures.

4. Reboot the system and issue **`ifconfig`** to verify (only the `eth2`, `eth3`, `bond0`, and `bond0_183` interfaces are shown here):

```
[root@ora-racb-1 ~]# ifconfig
bond0     Link encap:Ethernet  HWaddr 02:00:00:3F:FD:F3
          inet6 addr: fe80::ff:fe3f:fdf3/64 Scope:Link
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:8992  Metric:1
          RX packets:79 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:16217 (15.8 KiB)  TX bytes:4657 (4.5 KiB)

bond0_183 Link encap:Ethernet  HWaddr 02:00:00:3F:FD:F3
          inet addr:10.1.28.11  Bcast:10.1.28.15  Mask:255.255.255.248
          inet6 addr: fe80::ff:fe3f:fdf3/64 Scope:Link
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:8992  Metric:1
          RX packets:79 errors:0 dropped:0 overruns:0 frame:0
          TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:16217 (15.8 KiB)  TX bytes:3917 (3.8 KiB)

eth2      Link encap:Ethernet  HWaddr 02:00:00:3F:FD:F3
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:8992  Metric:1
          RX packets:34 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6899 (6.7 KiB)  TX bytes:2276 (2.2 KiB)

eth3      Link encap:Ethernet  HWaddr 02:00:00:3F:FD:F3
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:8992  Metric:1
          RX packets:45 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9318 (9.0 KiB)  TX bytes:2381 (2.3 KiB)
```

3.7.3 Setting up Oracle HAIP

Oracle introduced a new feature called HAIP. This feature allows for the setup of the private interconnect on RAC without the need for the user to create bonded interfaces. HAIP provides load balancing and HA across multiple network devices (up to four). You can enable HAIP during the Oracle installation process. HAIP requires two network interfaces from each node from the operating system. These interfaces incorporate VLAN tagging.

SUSE Linux Enterprise Server 11

Complete the following steps to set up HAIP in SUSE Linux Enterprise Server 11:

1. Add the two OSA card devices in the `/etc/udev/rules.d/` directory. The udev rules files are not shown. For more information, see "Setting up Ethernet bonding" on page 45.
2. Ensure that Layer2 is enabled for the Oracle RAC Interconnect.
3. Configure the two OSA card devices for VLAN tagging by using the `ifcfg` scripts. The `ifcfg` scripts can be found in the `/etc/sysconfig/network/` directory.
4. Reboot the system and run the **`ifconfig`** command.
5. Configure both OSA devices by using the ifcfg scripts. The following examples are `ifcfg` scripts for two OSA qeth devices that are bonded to a third interfaced named `bond0`. The interface names `eth1` and `eth2` were used. We reused the `eth1` interface name; therefore, we did not need an `eth3`. All IP information was removed from the `ifcfg-eth1`, `ifcfg-eth2`, and `ifcfg-bond0` files. VLAN tagging is added and a `bond0_182` is created. Separate IP information can be placed into the `ifcfg-bond0` script for non-VLAN tagging use. An MTU size of 8992 was used throughout the network (VSWITCH, OSA cards and network switch):

```
ora-raca-1:/etc/sysconfig/network # cat ifcfg-eth1
BOOTPROTO='static'
STARTMODE='onboot'
ETHTOOL_OPTIONS=''
INTERFACETYPE='qeth'
USERCONTROL='no'
MTU='8992'
NAME='OSA Express Network card (0.0.1303)'

ora-raca-1:/etc/sysconfig/network # cat ifcfg-eth1_182
BOOTPROTO='static'
ETHERDEVICE='eth1'
IPADDR='10.1.28.1/29'
NETMASK='255.255.255.248'
BROADCAST='10.1.28.7'
STARTMODE='auto'
MTU='8992'
NAME='OSA Express Network card (0.0.1303)'

ora-raca-1:/etc/sysconfig/network # cat ifcfg-eth2
BOOTPROTO='static'
STARTMODE='onboot'
ETHTOOL_OPTIONS=''
INTERFACETYPE='qeth'
USERCONTROL='no'
MTU='8992'
NAME='OSA Express Network card (0.0.1423)'
```

```
ora-raca-1:/etc/sysconfig/network # cat ifcfg-eth2_182
BOOTPROTO='static'
ETHERDEVICE='eth2'
IPADDR='10.1.28.3/29'
NETMASK='255.255.255.248'
BROADCAST='10.1.28.7'
STARTMODE='auto'
MTU='8992'
NAME='OSA Express Network card (0.0.1423)
```

6. Reboot the system and run the **ifconfig** command to verify (only the `eth1`, `eth2`, `eth1_182`, and `eth2_182` interfaces are shown here):

```
ora-raca-1:~ # ifconfig
eth1      Link encap:Ethernet  HWaddr 02:00:00:09:EC:87
          inet6 addr: fe80::ff:fe09:ec87/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:8992  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:1184 (1.1 Kb)

eth1_182  Link encap:Ethernet  HWaddr 02:00:00:09:EC:87
          inet addr:10.1.28.1  Bcast:10.1.28.7  Mask:255.255.255.248
          inet6 addr: fe80::ff:fe09:ec87/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:8992  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:578 (578.0 b)

eth2      Link encap:Ethernet  HWaddr 02:00:00:2A:C8:F4
          inet6 addr: fe80::ff:fe2a:c8f4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:8992  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:1254 (1.2 Kb)

eth2_182  Link encap:Ethernet  HWaddr 02:00:00:2A:C8:F4
          inet addr:10.1.28.3  Bcast:10.1.28.7  Mask:255.255.255.248
          inet6 addr: fe80::ff:fe2a:c8f4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:8992  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:578 (578.0 b)
```

3.8 Notes and observations

For VLAN tagging, we found that with RHEL6, we had to temporarily disable the Network Manager service to configure the VLAN tagging. Use the following **chkconfig** command to see if Network Manager is enabled:

```
[root@ora-racc-1 network-scripts]# chkconfig --list | grep NetworkManager
NetworkManager  0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

To disable or re-enable Network Manager, use the following command:

```
[root@ora-racc-1 network-scripts]# chkconfig NetworkManager off
[root@ora-racc-1 network-scripts]# chkconfig --list | grep NetworkManager
NetworkManager  0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

To change the MTU size, we had to try multiple times to increase MTU for `eth1_182` because it failed with the following message:

```
ora-raca-1:/etc/sysconfig/network # ifup  eth1_182
eth1_182  name: OSA Express Network card (0.0.1420)
RTNETLINK answers: Numerical result out of range
RTNETLINK answers: Numerical result out of range
Cannot set mtu of 8992 to interface eth1_182.
```

The following problems existed:

- We must make sure that our network router also was changed to accommodate the larger MTU size and Jumbo frames. All components in your network should have matching MTU sizes. This configuration prevents one component from attempting to auto-negotiate down to a lower MTU size.
- After this issue is addressed, ensure that you change the MTU size of the base `ethx` device before the VLAN tagged device is changed.

After the network router was set to an MTU of 8992, the interface `eth1` also can be set to MTU=8992. Also, if you are using VLAN tagging, change the VLAN tagged interface `eth1_xxx` to the appropriate MTU size.

3.9 Summary

The performance of each of these HA solutions is similar, including the operating system versions that were selected.

With all of these solutions, careful monitoring is needed from the Oracle AWR reports to the Linux **netstat** output to the z/VM network or HMC Network usages reports to ensure network bandwidths are not exceeded.

Figure 3-6 shows the Interconnect Ping latencies section of an Oracle AWR report when the performance of the Oracle RAC Interconnect is monitored.

```
Interconnect Ping Latency Stats            DB/Inst: RACC/racc1  Snaps: 393-394
-> Ping latency of the roundtrip of a message from this instance to -> target ir
-> The target instance is identified by an instance number.
-> Average and standard deviation of ping latency is given in miliseconds
-> for message sizes of 500 bytes and 8K.
-> Note that latency of a message from the instance to itself is used as
-> control, since message latency can include wait for CPU

   Target 500B Pin Avg Latency      Stddev  8K Ping Avg Latency      Stddev
 Instance    Count     500B msg    500B msg    Count       8K msg      8K msg
--------- -------- ----------- ----------- -------- ----------- -----------
        1       74         .19         .06       74         .17         .07
        2       74         .76        1.97       74         .94        1.94
          -------------------------------------------------------------
```

Figure 3-6 Oracle AWR Interconnect bandwidth

> **Important:** The decision to add more HA into your network IT solutions is based on your business requirements, hardware availability (OSA features), and the skill sets that are available.

VSWITCH aggregation, Linux Bonding, and Oracle's Redundant Interconnect have many benefits for providing more HA for running Oracle databases with Linux on System z workloads for planned or unplanned network outages.

4

Setting up SUSE Linux Enterprise Server 11 SP2 and Red Hat Enterprise Linux 6.2

This chapter provides an overview of the steps that are to prepare a Linux guest for the installation of a Single Instance Oracle 11gR2 database. It also includes information about both distributions.

The first step is to become familiar with the following Oracle support notes (which are available at `http://support.oracle.com`):

- Note 1306465.1: *Getting Started - 11gR2 Grid Infrastructure, ASM and DB (IBM: Linux on System z)*
- Note 1290644.1: *Requirements for Installing Oracle 11gR2 on SLES11 on IBM: Linux on System z (s390x) Also review note: OHASD fails to start on SuSE 11 SP2 on IBM: Linux on System z [ID 1476511.1]*
- Note 1470834.1: *Requirements for Installing Oracle 11gR2 on RHEL 6 on IBM: Linux on System z (s390x)*
- Note 1086769.1: *Ensure you have prerequisite rpms to install Oracle Database and AS10g(midtier) on IBM: Linux on System z (s390x)*

There are specific sections for SUSE Linux Enterprise Server 11 and Red Hat Enterprise Linux 6 and a common section that applies to both.

This chapter includes the following topics:

- Installing Oracle 11gR2 on SUSE Linux Enterprise Server guest
- Installing Oracle 11.2.0.3 on a Red Hat Enterprise Linux 6 guest
- Customization that is common to SUSE Linux Enterprise Server and Red Hat Enterprise Linux

4.1 Installing Oracle 11gR2 on SUSE Linux Enterprise Server guest

The minimum requirement to install Oracle 11gR2 is SUSE Linux Enterprise Server 10 SP3, or later. However, SUSE Linux Enterprise Server 11 SP2 (3.0.13-0.27) + is currently available and is preferable for 11gR2 because it incorporates support of various features of System z hardware. To verify your release, use the following command:

```
# cat /proc/version
Linux version 3.0.13-0.27-default (geeko@buildhost) (gcc version 4.3.4
[gcc-4_3-branch revision 152973] (SUSE Linux) ) #1 SMP Wed Feb 15 13:33:49 UTC
2012 (d73692b)
```

Tip: For SUSE Linux Enterprise Server 11 SP2, be sure to review the My Oracle Support note *OHASD fails to start on SuSE 11 SP2 on IBM: Linux on System z* [ID 1476511.1].

4.1.1 Linux required RPMs for SUSE Linux Enterprise Server 11

This section describes the following RPMs that are required to install Oracle on a SUSE Linux Enterprise Server 11 guests:

- Linux Base Installation

  When you are installing on a system, select the majority of the rpms that are required at installation time to help prevent issues with rpm dependencies. For example, selecting the “C” Libraries and functions contains the majority of the “C” library rpm packages.

- RPM checker

  You should use rpm checker to verify your system configuration before Oracle software is installed.

  Download the appropriate RPM checker from the bottom of the My Oracle Support (MOS) Note 1306465.1. The rpm checker checks that the required rpms for Oracle Grid and database installs. This prevents problems with the installation of Oracle. You must log on to the My Oracle Support website and select the SUSE Linux Enterprise Server 11 RPM checkers S11 Grid Infrastructure/Database RPM checker 11.2.0.2 (1.38 KB), which is found at this website:

  https://support.oracle.com/CSP/main/article?cmd=show&type=ATT&id=1086769.1:DB_S11_11202_ZLINUX

  Extract the download file and then install the extracted rpm to verify your Linux rpm requirements. The rpm checker does not actually install anything. Instead, the checker uses the dependencies of rpm to check your system, as shown in the following example:

```
# rpm -i ora-val-rpm-S11-DB-11.2.0.2-1.s390x.rpm
****************************************************************************
* Validation complete - Your sles11 OS has required rpms for Oracle 11.2   *
****************************************************************************
```

  Recommendation: Run the RPM checker command as the `root` user.

- Required RPMs

  The following rpm packages are required for each version of Linux. The RPM release numbers can be higher than the minimum versions listed here:

  Review the Note: 1383381.1 - 11.2.0.3 *PREREQ CHECK WARNING FOR MISSING compat-libstdc++-33.3.2.3-47.3 ON IBM: LINUX ON SYSTEM Z ON SLES 11*

  > **Important:** Certain packages require the 31 bit (s390) and the 64 bit version (s390x) of the rpm to be installed.

  The following packages should be installed as part of a base installation:

```
binutils-2.20.0-0.7.9.s390x.rpm
glibc-2.11.1-0.17.4.s390x.rpm
glibc-32bit-2.11.1-0.17.4.s390x.rpm
ksh-93t-9.9.8.s390x.rpm
libaio-0.3.109-0.1.46.s390x.rpm
libaio-32bit-0.3.109-0.1.46.s390x.rpm
libstdc++33-3.3.3-11.9.s390x.rpm
libstdc++33-32bit-3.3.3-11.9.s390x.rpm
libstdc++43-4.3.4_20091019-0.7.35.s390x.rpm
libstdc++43-32bit-4.3.4_20091019-0.7.35.s390x.rpm
libgcc43-4.3.4_20091019-0.7.35.s390x.rpm
make-3.81-128.20.s390x.rpm
```

  The remaining rpm requirements can be installed by selecting all the “C” Libraries and extensions or by manually installing each of the following rpms:

```
libaio-devel-0.3.109-0.1.46.s390x.rpm
libaio-devel-32bit-0.3.109-0.1.46.s390x.rpm
sysstat-8.1.5-7.9.56.s390x.rpm
glibc-devel-2.11.1-0.17.4.s390x.rpm (requires
linux-kernel-headers-2.6.32-1.4.13.noarch.rpm)
gcc-4.3-62.198.s390x.rpm (requires gcc43-4.3.4_20091019-0.7.35.s390x.rpm)
glibc-devel-32bit-2.11.1-0.17.4.s390x.rpm
gcc-32bit-4.3-62.198.s390x.rpm (requires
gcc43-32bit-4.3.4_20091019-0.7.35.s390x.rpm and
libgomp43-32bit-4.3.4_20091019-0.7.35.s390x.rpm)
libstdc++43-devel-4.3.4_20091019-0.7.35.s390x.rpm
gcc-c++-4.3-62.198.s390x.rpm (requires
gcc43-c++-4.3.4_20091019-0.7.35.s390x.rpm)
libstdc++43-devel-32bit-4.3.4_20091019-0.7.35.s390x.rpm
libstdc++-devel-4.3-62.198.s390x.rpm
libcap1-1.10-6.10.s390x.rpm
```

  The following rpm command is used to verify the full extensions of the rpms. Some of the requirements need the s390 (31 bit), and some need the s390x (64 bit) version of the rpm:

```
# rpm -qa --queryformat="%{n}-%{v}-%{r}.%{arch}.rpm" | grep <package>
```

4.1.2 Network Time Protocol TIME option

If you are performing an Oracle Grid/Automated Storage Manager (ASM) installation, the grid install performs a system check to verify that the Cluster Time Synchronization Service is set to prevent the system time from being adjusted backward.

If you are installing Oracle Grid for Single Instance ASM, or Oracle RAC you should modify the NTP configuration to include the `"slueing:` option with the ***-x*** parameter.

Edit the `/etc/sysconfig/ntp` file and add the ***-x*** flag, as shown in Example 4-1.

Example 4-1 Modifying the NTP configuration

```
#NTPD_OPTIONS="-g -u ntp:ntp"
NTPD_OPTIONS="-x -g -u ntp:ntp"
```

Restart the network time protocol daemon after you complete this task by running the following command as the root user:

```
# /sbin/service ntp restart
Shutting down network time protocol daemon (NTPD)                    done
Starting network time protocol daemon (NTPD)                         done
# ps -ef | grep ntp | grep -v grep
ntp  56945 1  0 11:06 00:00:00 /usr/sbin/ntpd -p /var/run/ntp/ntpd.pid -x -g -u
ntp:ntp -i /var/lib/ntp -c /etc/ntp.conf
```

Next, we must configure the system by using the following command so that the NTP daemon is started on reboot:

```
# chkconfig --level 35 ntp on
```

You might encounter the problem that is shown in Example 4-2 when Oracle runs its system pre-checks.

Example 4-2 Clock synchronization error

```
PRVE-0029 : Hardware clock synchronization check could not run on node xxxxx"
```

To resolve this problem, add the following lines to the `/etc/init.d/halt.local` file:

```
CLOCKFLAGS="$CLOCKFLAGS --systohc"
#/sbin/hwclock --systohc
```

You can now proceed to 4.3, “Customization that is common to SUSE Linux Enterprise Server and Red Hat Enterprise Linux” on page 60.

4.2 Installing Oracle 11.2.0.3 on a Red Hat Enterprise Linux 6 guest

The section describes the process that is used to install Oracle 11g R2 on a Red Hat Enterprise Linux 6 (RHEL 6) guest only.

If you are using SUSE Linux Enterprise Server, see 4.1, “Installing Oracle 11gR2 on SUSE Linux Enterprise Server guest” on page 56.

For more information about how to install Red Hat Enterprise Linux 6 for an Oracle Database, see Appendix A, “Setting up Red Hat Enterprise Linux 6.3 for Oracle” on page 311.

For Oracle Database 11gR2 (11.2.0.3), the minimum version is RHEL 6.2, kernel -2.6.32-220 or higher. This was certified in Q1 2013.

To check the version of RHEL you have installed, use the following command:

```
# cat /proc/version
Linux version 2.6.32-220.el6.s390x (mockbuild@s390-001.build.bos.redhat.com)
(gcc version 4.4.5 20110214 (Red Hat 4.4.5-6) (GCC) ) #1 SMP Wed Nov 9 08:20:08
EST 2011.
```

You also should review the following notes:

- Note 1377392.1: *How to Manually Configure Disk Storage devices for use with Oracle ASM 11.2 on IBM: Linux on System z Red Hat 6*
- Note 1459030.1: 11.2.0.3 *Grid Installer Hangs at 75% When Using DASD Softlink Device*
- Note 1514012.1: *runcluvfy stage -pre crsinst generates reference data is not available for verifying prerequisites for RHEL 6*.

4.2.1 Verify SELinux is permissive or disabled

Oracle 11gR2 on an RHEL 6 system must have SELinux disabled or set to permissive or `sqlplus` does not work properly. To verify your SELinux setting, check the `/etc/selinux/config` file and ensure the SELinux setting is not set to Enforcing; otherwise, specify `SELINUX=permissive` and reboot:

```
# cat /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - SELinux is fully disabled.
SELINUX=permissive
# SELINUXTYPE= type of policy in use. Possible values are:
#       targeted - Only targeted network daemons are protected.
#       strict - Full SELinux protection.
SELINUXTYPE=targeted
```

To change dynamically, you can run the **getenforce** command as `root` to change the SELinux security mode, as shown in the following example:

```
# getenforce     (returns "Enforcing")
# setenforce 0
# getenforce     (returns "Permissive")
```

Important: Disabling SELinux or setting it to “permissive” mode can have security considerations. It is possible to keep SELinux enabled and add manual exclusion rules. For more information, see Oracle note [457458.1], *How to disable or set SELinux to permissive mode*.

4.2.2 Linux required RPMs for Red Hat Enterprise Linux installations

For more information about how to set up a Linux guest with all the required rpms for an Oracle Database installation, see Appendix A, “Setting up Red Hat Enterprise Linux 6.3 for Oracle” on page 311.

The rpm checker for RHEL 6 is `ora-val-rpm-EL6-DB-11.2.0.3-1.s390x.rpm`.

4.2.3 Setting NTP TIME for Red Hat Enterprise Linux (optional only for Oracle Grid installations)

Oracle Grid/ASM performs a system check to verify that the Cluster Time Synchronization Service is set in such a way as to prevent the system time from being adjusted backward.

If you are installing Oracle Grid for Single Instance ASM or Oracle RAC, you should modify the NTP configuration to include the `slewing` option with the ***–x*** parameter.

To do this on RHEL, edit the `/etc/sysconfig/ntpd` file and add the ***-x*** flag, as shown in the following example:

```
# cat /etc/sysconfig/ntpd
# Drop root to id 'ntp:ntp' by default.
OPTIONS="-x -u ntp:ntp -p /var/run/ntpd.pid"
# Set to 'yes' to sync hw clock after successful ntpdate
SYNC_HWCLOCK=no
# Additional options for ntpdate
NTPDATE_OPTIONS=""
```

Restart the network time protocol daemon after you complete this task as the root user with the following command:

```
# /sbin/service ntpd restart
```

Next, configure the system so that the NTP daemon is started on reboot by using the following command:

```
# chkconfig --level 35 ntpd on
```

Proceed to 4.3, “Customization that is common to SUSE Linux Enterprise Server and Red Hat Enterprise Linux”.

4.3 Customization that is common to SUSE Linux Enterprise Server and Red Hat Enterprise Linux

This section describes the following customization that must be done to the Linux guest that is common to SUSE Linux Enterprise Server 11 and Red Hat Enterprise Linux 6:

- Kernel parameters
- Oracle User Groups and accounts
- File Descriptor limits
- User directories
- Other RPMs

4.3.1 Required parameters for Oracle

This section describes the required parameters for Oracle.

Kernel parameters

As the `root` user, ensure that the required Kernel parameters are set in `/etc/sysctl.conf` file, as shown on Example 4-3. The recommended Kernel requirements are listed in My Oracle Support notes, such as *Requirements for Installing Oracle 11gR2 on SLES 11 on IBM: Linux on System z (s390x)*, ID 1290644.1 and *Requirements for Installing Oracle 11.2.0.3 RDBMS on RHEL 6 on IBM: Linux on System z (s390x)*, ID 1470834.1.

Example 4-3 Sample /etc/sysctl.conf

```
# Oracle Kernel Specific parameters
#
#fs.file-max = 512 x oracle processes (for example 6815744 for 13312 processes)
fs.file-max = 6815744
# fs.aio-max-nr = 3145728 (use for really large concurrent I/O databases
fs.aio-max-nr = 1048576
#kernel.shmall = set to (sum of all sga's on  system)  / 4096   or  a default of
2097152
kernel.shmall = 2097152
#kernel.shmmax=MAX (1/2 the virtual RAM , largest SGA_MAX_SIZE/SGA_TARGET on
system)
kernel.shmmax =  4218210304
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
kernel.spin_retry = 2000
#vm.nr_hugepages = 4000 (Use for large SGA's > 10 GB)
```

Complete the following steps to change these values:

1. Make a copy of the `/etc/sysctl.conf` files if you are making any changes. Use the **-p** option to preserve the date, as shown in the following example:

   `# cp -p  /etc/sysctl.conf  /etc/sysctl.conf.old`
2. Edit the `/etc/sysctl.conf` file.
3. Run the **sysctl -p** command for the kernel parameter changes to take effect.

Network configuration

You should comment out any IPV6 (see number 2 in Example 4-4) entries from your `/etc/hosts` file if you are not using IPv6 IP addresses. Also, the first line of the `/etc/hosts` should contain local hosts, as shown in number 1 in Example 4-4.

Example 4-4 The hosts file in the lab environment.

```
# cat /etc/hosts
127.0.0.1 localhost.localdomain localhost 1
# special IPv6 addresses
#localhost ipv6-localhost ipv6-loopback 2
```

```
9.82.34.164            ora1.wsclab.washington.ibm.com  ora1
# Additional Required Only for Oracle RAC install
9.82.34.165            ora2.wsclab.washington.ibm.com  ora2
10.0.0.164             ora1-priv.wsclab.washington.ibm.com ora1-priv
10.0.0.165             ora2-priv.wsclab.washington.ibm.com ora2-priv
9.82.34.167            ora1-vip.wsclab.washington.ibm.com  ora1-vip
9.82.34.168            ora2-vip.wsclab.washington.ibm.com ora2-vip
#
#  If Not using Oracle SCAN IP's for Oracle then setup 2 DNS entries as below
#
#9.82.34.166          ora-cluster.wsclab.washington.ibm.com ora-cluster crs
#9.82.34.169          ora-cluster-scan.wsclab.washington.ibm.com ora-cluster-scan
9.82.34.167            ora1-vip.wsclab.washington.ibm.com  ora1-vip
9.82.34.168            ora2-vip.wsclab.washington.ibm.com
ora2-vip
```

Oracle also requires that the host name be the fully qualified domain name, with a corresponding entry in the `/etc/hosts` file, as shown in the following example:

```
# hostname
ora1.wsclab.washington.ibm.com
```

4.3.2 Oracle RAC installations only

For Oracle RAC installations, more IP addresses are required (see Example 4-5) for the other nodes in the RAC cluster (`ora2`), the Private Interconnect (-priv), and the variable IPs (-vip), which are created when the Oracle Grid starts.

Example 4-5 Other IP addresses identified in the lab environment

```
10.0.0.164             ora1-priv.wsclab.washington.ibm.com ora1-priv
10.0.0.165             ora2-priv.wsclab.washington.ibm.com ora2-priv
```

Tip: For other steps and requirements, see Chapter 3, “Network connectivity options for Oracle on Linux on IBM System z” on page 29 and Appendix B in the *Installing Oracle 11gR2 RAC on Linux on System z*, REDP4788.

One other network interface on each server must be created, such as `hsi0` - virtual HiperSocket. This network interface is between the Linux Guests for Oracle’s Interconnect and should be on a private non-routable interface (`192.x.x.x or 10.x.x.x`). Only the nodes in the RAC cluster should contact the private interface.

You also require two other IP addresses for the Oracle Virtual IPs (VIPs) that must be on the same subnet as the public `eth0` interface, as shown in Example 4-6.

Example 4-6 Oracle VIPs

```
9.82.34.167            ora1-vip.wsclab.washington.ibm.com  ora1-vip
9.82.34.168            ora2-vip.wsclab.washington.ibm.com ora2-vip
```

Finally, you need three SCAN IP addresses to be defined as Class A DNS entries, as shown in Example 4-7 (there are three IP addresses for each DNS entry) for the new 11gR2 Oracle RAC systems. These should also be on the same subnet as the public interface.

Example 4-7 DNS SCAN entries

```
rac-scan IN A 9.82.34.166
rac-scan IN A 9.82.34.167
rac-scan IN A 9.82.34.168

# Note 3 IPs to one DNS (host file entry, but we require DNS entries for this to
work)
#
9.82.34.166rac-scan.<domain name>rac-scan
9.82.34.167rac-scan.<domain name>rac-scan
9.82.34.168rac-scan.<domain name>rac-scan
```

If you cannot set up DNS SCAN entries at this time, you can define two `/etc/host` entries on each of the nodes, but you receive a warning that can be ignored during the installation, as shown in Example 4-8.

Example 4-8 Non-SCAN Oracle RAC configuration

```
9.82.34.166          ora-cluster.wsclab.washington.ibm.com ora-cluster crs
9.82.34.169          ora-cluster-scan.wsclab.washington.ibm.com ora-cluster-scan
```

> **Important:** When the two network interfaces for Oracle RAC are configured (public and private interfaces), you must have ARP enabled (that is, NOARP must not be configured). The `root.sh` script fails on the first node if NOARP is configured.

Example 4-9 shows the command **`ifconfig -a`** run as though root user.

Example 4-9 ifconfig output

```
eth0      Link encap:Ethernet  HWaddr 02:00:00:00:00:05
          inet addr: 9.82.34.164  Bcast: 9.82.63.255 Mask:255.255.255.224
          inet6 addr: fe80::200:0:100:5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1492  Metric:1
          RX packets:6503 errors:0 dropped:0 overruns:0 frame:0
          TX packets:199 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:697138 (680.7 Kb)  TX bytes:24804 (24.2 Kb)

hsi0      Link encap:Ethernet  HWaddr 02:00:00:00:00:06
          inet addr:10.0.0.164  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::ff:fe00:6/64 Scope:Link
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:8192  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:390 (390.0 b)
```

4.3.3 Create and verify required UNIX groups and Oracle user accounts

When Oracle 11gR2 is installed, Oracle recommends that two groups be created: one for the group named `dba`, and another group called `oinstall`. (It is possible to install with one group; for example, `dba`.)

If only database executable files are installed, often one user account called `oracle` is created.

If Oracle Grid for Oracle ASM or a Real Application Cluster (RAC) system is installed, another user account called `grid` should be created to manage the grid infrastructure components.

As part of a grid infrastructure installation, Oracle changes certain directories and files to have `root` access privileges. Having separate user IDs (one for `grid`, and one for `oracle`) makes it easier to configure the environment variables that are required to maintain each environment.

To verify that the Linux groups and users were created, you can view the group and password files by using the following commands:

```
# cat /etc/passwd | grep oracle
# cat /etc/group
```

If your users and groups were not created, run the commands that are shown in Example 4-10 to create users. Having consistent group IDs (that is, 501) and user IDs (for example, 502) across nodes is required, particularly if you share storage or files between systems.

Example 4-10 Commands to create users

```
/usr/sbin/groupadd -g 501 oinstall
/usr/sbin/groupadd -g 502 dba
/usr/sbin/useradd -u 501 -g oinstall -G dba -s /bin/ksh -m grid
/usr/sbin/useradd -u 502 -g oinstall -G dba -s /bin/ksh -m oracle
```

Use the following commands to set the passwords for the `grid` and `oracle` users:

```
# passwd grid
# passwd oracle
```

Tip: If you encounter INS-06101 errors, verify that the host name is correct in the `/etc/hosts` file and host name is fully qualified.

4.3.4 Setting file descriptors limits for the oracle and grid users

As the `root` user, edit or verify the `/etc/security/limits.conf` file. If you created a separate user for the Oracle Grid user, the file descriptor limit or ulimit entries for the grid user should be created, as shown in Example 4-11.

Example 4-11 Unlimit entries

```
#vi  /etc/security/limits.conf
grid            soft            nofile  1024
grid            hard            nofile  65536
grid            soft            nproc   2047
grid            hard            nproc   16384
#
oracle          soft            nofile  1024
oracle          hard            nofile  65536
oracle          soft            nproc   2047
oracle          hard            nproc   16384
#
# Use memlock for Huge Pages  support (commented out)
#*              soft            memlock   3145728
#*              hard            memlock   3145728
```

Ensure that the `/etc/pam.d/login` file has an entry for `pam_limits.so`. Also, you should make a backup if changes are made to `/etc/pam.d/login` and to test any changes with a superuser or login before logging off, as a typographical error can make future logins problematic, as shown in Example 4-12.

Example 4-12 Making a backup and verification

```
# cp /etc/pam.d/login /etc/pam.d/login.old
# cat /etc/pam.d/login
#%PAM-1.0
auth     required       pam_nologin.so
…
session  optional       pam_mail.so standard
session  required       /lib/security/pam_limits.so
session  required       pam_limits.so
session  optional       pam_mail.so standard
```

To increase the limits at `oracle` logon, as the Oracle User, verify the `oracle` user's profile (for example, `/home/oracle/.profile` for ksh users) and ensure the following lines were added:

```
#vi .profile
ulimit -n 65536
ulimit -u 16384
```

Another method is to modify the main system profile by adding the following lines to the file called `/etc/profile`, as shown on Example 4-13. Change this if the `oracle` user is using a separate user shell program, such as csh or bash.

Example 4-13 Modifying the main system profile

```
If [ $USER = "oracle" ]; then
  ulimit -u 16384
  ulimit -n 65536
fi
# Optional Grid User
If [ $USER = "grid" ]; then
  ulimit -u 16384
  ulimit -n 65536
fi
```

You should then log in as the `oracle` user and run the command **`ulimit -a`** to verify everything is configured correctly, as shown in Example 4-14.

Example 4-14 Verifying the configuration

```
#  su - oracle
$  ulimit -a
address space limit (kbytes)   (-M)  unlimited
core file size (blocks)        (-c)  0
cpu time (seconds)             (-t)  unlimited
data size (kbytes)             (-d)  unlimited
file size (blocks)             (-f)  unlimited
locks                          (-L)  unlimited
locked address space (kbytes)  (-l)  unlimited
nice                           (-e)  0
nofile                         (-n)  65536
nproc                          (-u)  16384
pipe buffer size (bytes)       (-p)  4096
resident set size (kbytes)     (-m)  unlimited
rtprio                         (-r)  0
socket buffer size (bytes)     (-b)  4096
stack size (kbytes)            (-s)  10240
threads                        (-T)  not supported
process size (kbytes)          (-v)  unlimited
```

4.3.5 Pre-create user directories for product installs

If you are installing Oracle Grid, the `GRID_BASE` directory must be different from the grid's product directory (where the executable files are installed). In Example 4-15, `/oracle` is the logical volume for installing the Oracle executable files.

Example 4-15 Creating directories for Oracle Grid

```
#  Make Directories for Oracle Grid
mkdir -p /u01/grid/base
mkdir -p /u01/grid/11.2
chown -R grid:oinstall /u01/grid
chmod -R 775 /u01/grid
#  Make Directories for Oracle User (database)
mkdir -p /u01/oracle/11.2
chown -R oracle:oinstall /u01/oracle
chmod -R 775 /u01/oracle
```

Do not set any of the standard environment variables. If standard environment variables are already set, unset the standard `ORACLE_HOME` and `LD_LIBRARY_PATH` with a comment, as shown in lines 1 and 2 in Example 4-16 and Example 4-17.

Example 4-16 Example of /home/grid/.profile

```
$ cat .profile
export ORACLE_BASE=/u01/grid/base
export GRID_BASE=/u01/grid
#export ORACLE_HOME=$GRID_BASE/11.2
#
# comment out the following lines for use later,  do not have set for runInstaller
#
#export PATH=$ORACLE_HOME/bin:$ORACLE_HOME/OPatch:$PATH:. 1
#export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH 2
umask 022
#defaults for shell startup for ulimits of oracle user
ulimit -u 16384
ulimit -n 65536
```

Example 4-17 Example of /home/oracle/.profile

```
$cat .profile

export ORACLE_BASE=/u01/oracle
#export ORACLE_HOME=$ORACLE_BASE/11.2
#
# comment out the following lines for use later,  do not have set for runInstaller
#
#export PATH=$ORACLE_HOME/bin:$ORACLE_HOME/OPatch:$PATH:. 1
#export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH 2
umask 022
#defaults for shell startup for ulimits of oracle user
ulimit -u 16384
ulimit -n 65536
```

4.3.6 Other rpm for grid installs for SUSE Linux Enterprise Server and Red Hat Enterprise Linux

If you are performing an Oracle RAC install, you must install the `cvudisk-1.0.9-1` rpm package from the Oracle 11gR2 distribution media, as shown in Figure 4-1.

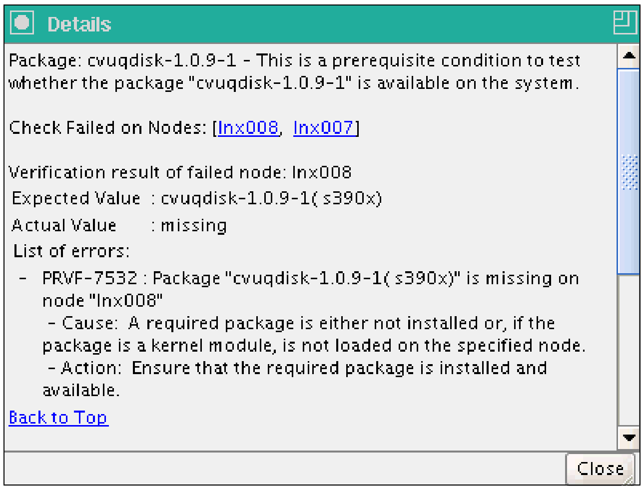


Figure 4-1 Installing other RPMs

You can create a fix up script or install the RPM from the software distribution on each of the nodes in the RAC cluster.

To run the `fixup` scripts, complete the following steps:

1. Open a terminal window.
2. Log in as the `root` user.
3. Run the scripts.
4. Return to the window that is shown in Figure 4-2 and click **OK**.

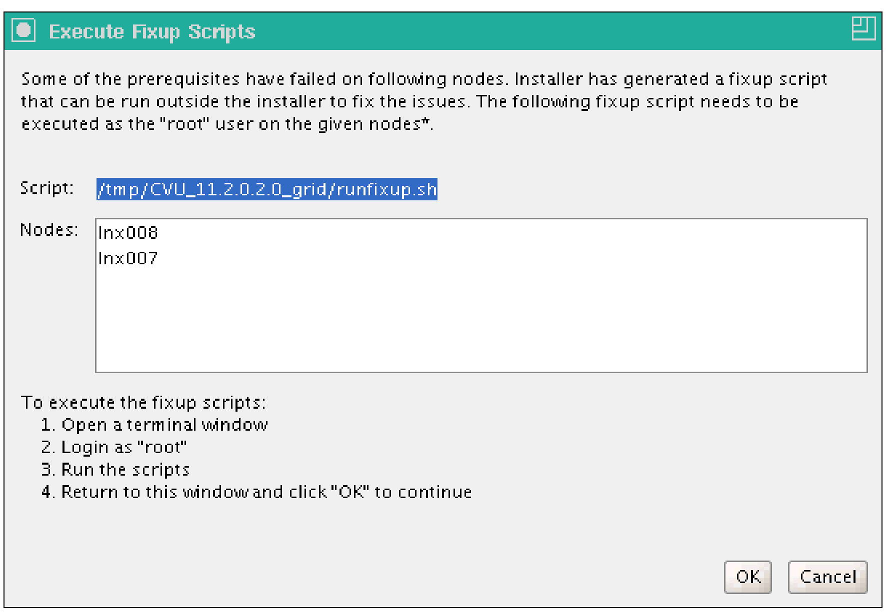


Figure 4-2 Running fixup scripts

Example 4-18 shows running the fixup script.

Example 4-18 Running the fixup script

```
root@lnx007 CVU_11.2.0.2.0_grid]# ./runfixup.sh
/usr/bin/id
Response file being used is :./fixup.response
Enable file being used is :./fixup.enable
Log file location: ./orarun.log
Installing Package /tmp/CVU_11.2.0.2.0_grid//cvuqdisk-1.0.9-1.rpm
Preparing...                ########################################### [100%]
   1:cvuqdisk               ########################################### [100%]
```

The RPM can also be found in the `<Grid CD>/Disk1/rpm/cvuqdisk-1.0.9-1.rpm` directory and installed in advance.

If you are ready to install an Oracle Single Instance database, see Appendix B, “Installing Oracle and creating a database 11.2.0.3 on Red Hat Enterprise Linux 6” on page 335.

5

Using the Cloud Control agent to manage Oracle databases

This chapter provides information about how to deploy Oracle Enterprise Manager Cloud Control 12c Release 2 (12.1.0.2) agents on a Linux on System z environment to manage Oracle databases.

To manage Oracle databases, we can use Database Control or Cloud Control. Both are intuitive web-based tools from Oracle. Database Control is installed with every database installation and can be used to monitor a single database instance or a clustered database, whereas the Cloud Control server can monitor many databases that are running on multiple environments from a single console. In addition to databases monitoring, other applications such as Siebel, PeopleSoft, and support for monitoring certain non-Oracle products, IBM WebSphere® Application Server, can also be monitored from a single Cloud Control console. While the Cloud Control Server does not run on Linux on System z, the agents are running on the Linux on System z guests and they communicate with the Cloud Control Server.

In this chapter, we share our experiences with installing a Cloud Control Server on x86 based Linux server and how the agents can be deployed from there to monitor the databases that are running on Linux on System z. Before deploying the agents, the Cloud Control Software Library must be updated with the required levels of agent software and plug-ins. This operation must be performed by connecting to Oracle repository site online. Starting with Oracle Enterprise Manager Cloud Control 12c Release 2 (12.1.0.2), command line utility **`emcli`** can also be used to update the Grid Control Software Library in offline mode. Then, the agents can be deployed from the Cloud Control Console and by using silent agent deployment option at the Linux guests (these procedures are provided in this chapter).

This chapter includes the following topics:

- Basic Enterprise Manager Cloud Control Architecture
- Creating Cloud Control infrastructure on x86 Linux
- Updating the Cloud Control Software Library in online mode
- Updating the Cloud Control Software Library in offline mode
- Deploying the Agents from Cloud Control console
- Deploying Management Agents in silent mode
- Adding the databases for monitoring
- Summary

5.1 Basic Enterprise Manager Cloud Control Architecture

The Cloud Control Architecture comprises the following components:

- Oracle Management Services (OMS)

  The OMS is a web-based application that coordinates with the Management Agents and the plug-ins and stores the collected information in a repository. It also renders the user interface for Enterprise Manager Cloud Control. The OMS is deployed to the Oracle Middleware home (Middleware home) and installs Oracle WebLogic Server if it does not exist in the environment.

- Oracle Management Repository

  At the time of Enterprise Manager Cloud Control installation, the Management Repository is configured in the existing database. This is where all the information that is collected by the Management Agent is stored.

- Plug-ins

  Plug-ins are deployed to the OMS and the Management Agent. They work in with the OMS and Management Agent to monitor all targets in the environment. By default, the plug-ins that are needed for Oracle Databases are installed automatically at the time of Enterprise Manager Cloud Control installation.

- Oracle Management Agent

  The Management Agent is deployed to the hosts, which are monitored by the Enterprise Manager Cloud Control and works with the plug-ins to monitor the targets that are running on that managed host.

- Cloud Console

  Cloud Console is the centralized location to monitor and manage the systems and services.

Figure 5-1 shows the Cloud Control Architecture as shown in the *Oracle Enterprise Manager Grid Control Basic Installation Guide*.

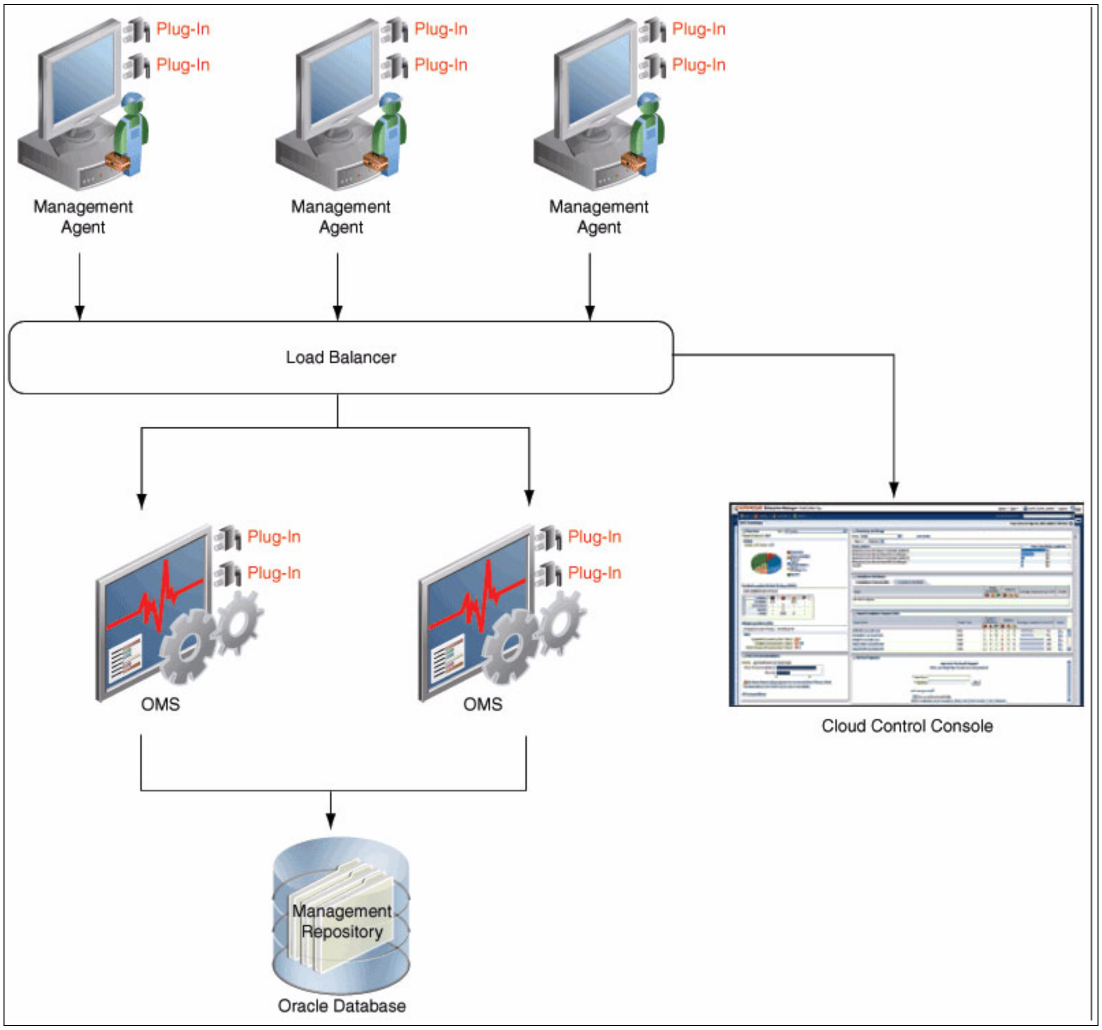


Figure 5-1 Cloud Control Architecture

5.2 Creating Cloud Control infrastructure on x86 Linux

For more information, see the My Oracle support document FAQ: *Enterprise Manager Cloud Control 12c Availability / Certification / Install / Upgrade Frequently Asked Questions* [ID 1363863.1].

In our environment, we created Oracle Enterprise Manager Cloud Control on a x86-64 Red Hat Enterprise Linux 6 (RHEL 6.2) Linux Server.

For more information about the hardware and software requirements for the server architecture where the Cloud Control is installed, see *Oracle Enterprise Manager Cloud Control Basic Installation Guide, 12c Release 2 (12.1.0.2)* E22624-15.

We followed the basic processes to install 11.2.0.3 Database Enterprise Edition and created a database. This installation procedure is beyond the scope of this document. For more information, see the Oracle documentation.

In the server, we created the Oracle Enterprise Manager Cloud Control infrastructure by following these steps:

1. Download and extract the installation `.zip` files.
2. Start the installation in a graphical terminal, such as VNC client.
3. Customize the Enterprise Manager Cloud Control during installation.
4. Customize the Oracle database as per repository configuration requirements.
5. Run the **`root.sh`** command.
6. Make sure that Enterprise Manager Cloud Control Console is accessible.

5.2.1 Downloading and extracting the installation files

The following files are downloaded from Oracle OTN network (we confirmed that we were authorized to download the software) for Enterprise Manager Cloud Control 12c Release 2 (12.1.0.2) for Linux x86-64:

- `em12cr2_linux64_disk1.zip (1,604,870,804 bytes) (cksum - 1628274676)`
- `em12cr2_linux64_disk2.zip (1,685,863,807 bytes) (cksum - 1265572904)`
- `em12cr2_linux64_disk3.zip (2,290,032,132 bytes) (cksum - 2732732695)`

We extracted the files and the following directory structure was created:

```
drwxrwxrwx. 10 oracle oinstall       4096 Oct 11 08:33 ./
drwxrwxrwx.  5 oracle oinstall       4096 Oct 11 08:34 ../
drwxrwxr-x.  8 oracle oinstall       4096 Aug 31 17:02 install/
drwxrwxr-x.  4 oracle oinstall       4096 Aug 31 17:01 jdk/
drwxrwxr-x.  4 oracle oinstall       4096 Aug 31 17:03 libskgxn/
drwxrwxr-x.  4 oracle oinstall       4096 Aug 31 16:52 oms/
drwxrwxr-x.  2 oracle oinstall       4096 Aug 31 17:04 plugins/
-rwxrwxr-x.  1 oracle oinstall      99186 Sep  4 12:49 Release_Notes.pdf*
drwxrwxr-x.  2 oracle oinstall       4096 Aug 31 17:02 response/
-rwxrwxr-x.  1 oracle oinstall       5136 Feb 10  2010 runInstaller*
drwxrwxr-x.  9 oracle oinstall       4096 Aug 31 17:03 stage/
drwxrwxr-x.  2 oracle oinstall       4096 Aug 31 17:02 wls/
-rwxrwxr-x.  1 oracle oinstall 1510253719 Aug 31 16:52 WT.zip*
```

5.2.2 Installing and configuring the Enterprise Manager Cloud Control 12c

The following tasks are completed by the installation wizard as part of a new Enterprise Manager system:

- Install the Middleware Components in the Middleware home (in our example, `/u01/app/mw`). The following components are installed in the Middleware home:
  - Java Development Kit (JDK) 1.6 v24
  - Oracle WebLogic Server 11g Release 1 (10.3.5)
  - Oracle Management Service 12c Release 2 (12.1.0.2)
  - Oracle JRF 11g Release (11.1.1.6.0), which includes `oracle_common` directory
  - Oracle Web Tier 11g Release (11.1.1.6.0), which includes `Oracle_WT` directory
  - Oracle plug-ins
- Oracle Management Agent 12c Release 2 (12.1.0.2) is installed in the agent base directory that is specified during installation (outside the Middleware home, in our example, `/u01/app/oracle/agent`).
- An Oracle WebLogic domain called `GCDomain` and a default user account, `weblogic` were used as the administrative user and a node manager account also was created.

- Oracle Management Service is configured in the Instance Base location (`gc_inst`) in the Middleware home for storing all configuration details that are related to Oracle Management Service 12c (in our example, `/u01/app/mw/gc_inst`).
- Configured Oracle Management Repository in the existing Oracle Database (in our example, Oracle SID: `orcl`).
- Configured the various installed components.

Start the installation wizard as `oracle` user from the extracted directory location of the downloaded installation files. As shown in Figure 5-2, the installation starts to specify the Oracle Support Credentials. (The Software Updates option is skipped in Figure 5-2.)

Figure 5-2 My Oracle Support Details window

During prerequisite checks, the installer verifies the requirements for installation. We made sure that all the steps are successful.

However, during software packages checks, we encountered the following software error, as shown in Figure 5-3:

```
Checking for glibc-devel-2.5-49.i386; Not found. failed <<<<
```

In SUSE Linux Enterprise Server 11, the following package also failed:

```
Checking for libstdc++43-4.3; Not found. Failed <<<<
```

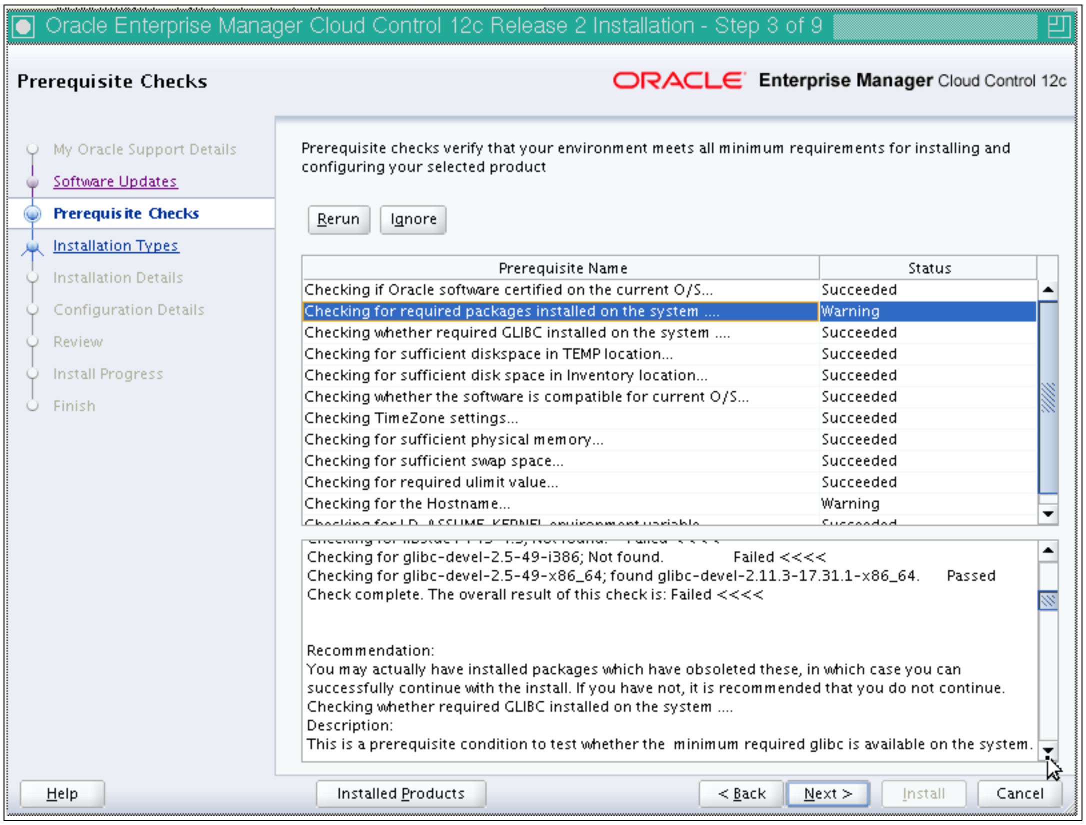


Figure 5-3 Prerequisite checks

However, the warnings that are shown in Figure 5-3 can be ignored for the following packages if you have them already installed in your system as per My Oracle Support documents.

Recommended My Oracle Support documents: For more information, see the following documents:

- EM 12c: *Installation on OEL6 64-bit Fails At Pre-requisite Check Due To Missing Package 'glibc-devel-2.5-49.i386'*, ID 1478035.1
- EM 12c: *Agent Installation on SLES11 fails at Pre-requisite check “Checking for libstdc++-4.1.0; Not found. Failed*,” ID 1471398.1

Ignore the warnings and continue the installation.

As shown in Figure 5-4, we selected **Create a New Enterprise System** and the **Advanced** option.

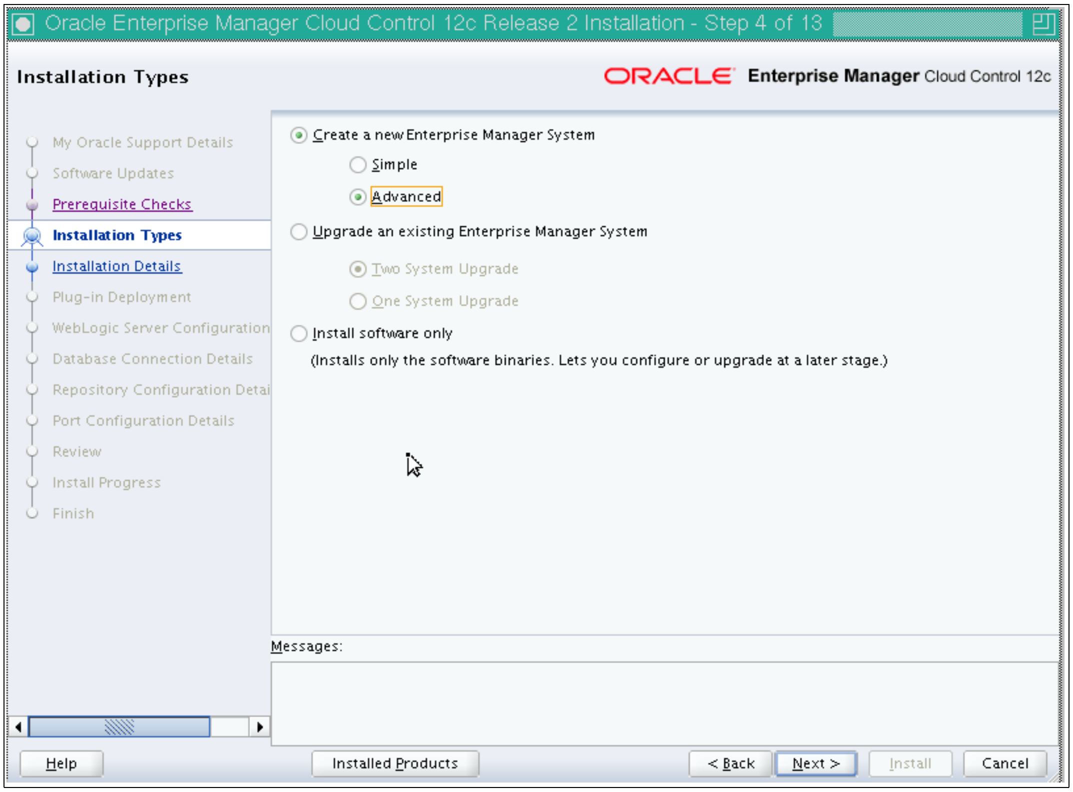


Figure 5-4 Installation types

As shown in Figure 5-5, we specified the middleware home location, agent base directory location, and the host name where the installation is carried out.

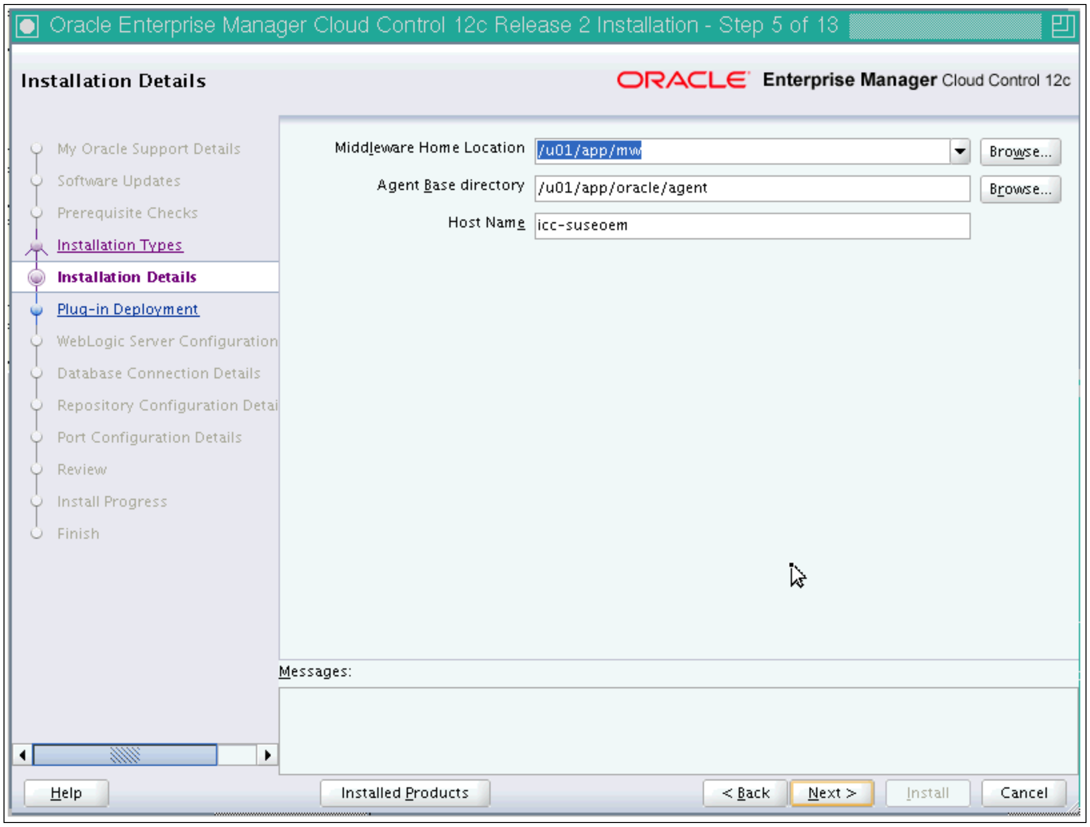


Figure 5-5 Installation Details

As shown in Figure 5-6, the mandatory plug-ins are automatically grayed out and we can select any other plug-ins that are needed. In our example, we left the default selection.

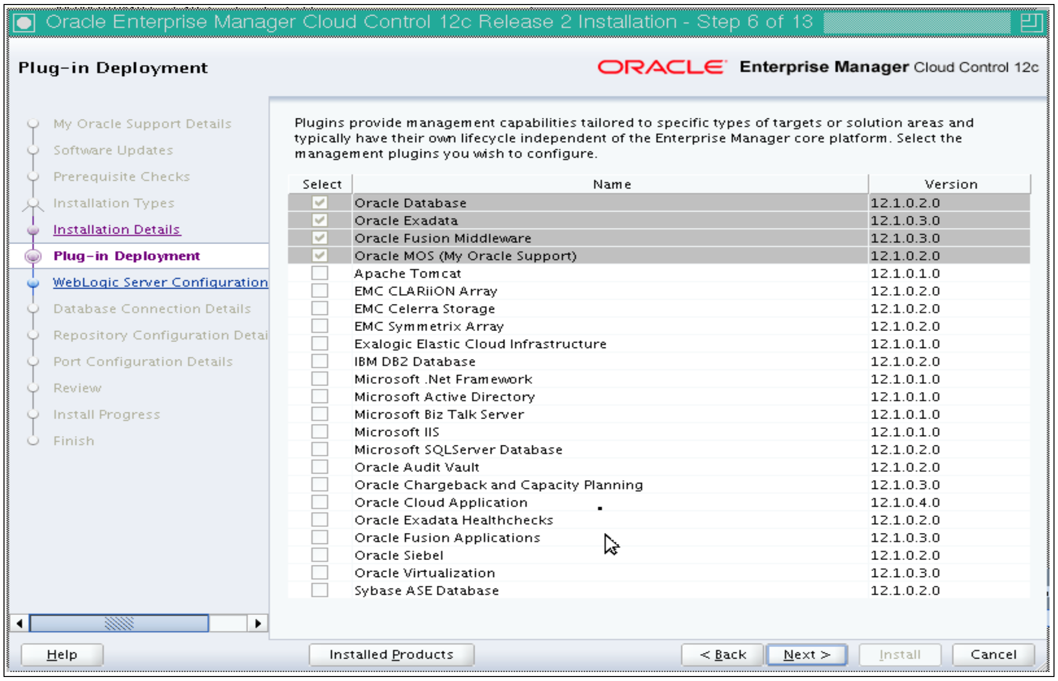


Figure 5-6 Plug-in Deployment window

As shown in Figure 5-7, we specified the information for WebLogic configuration requirements.

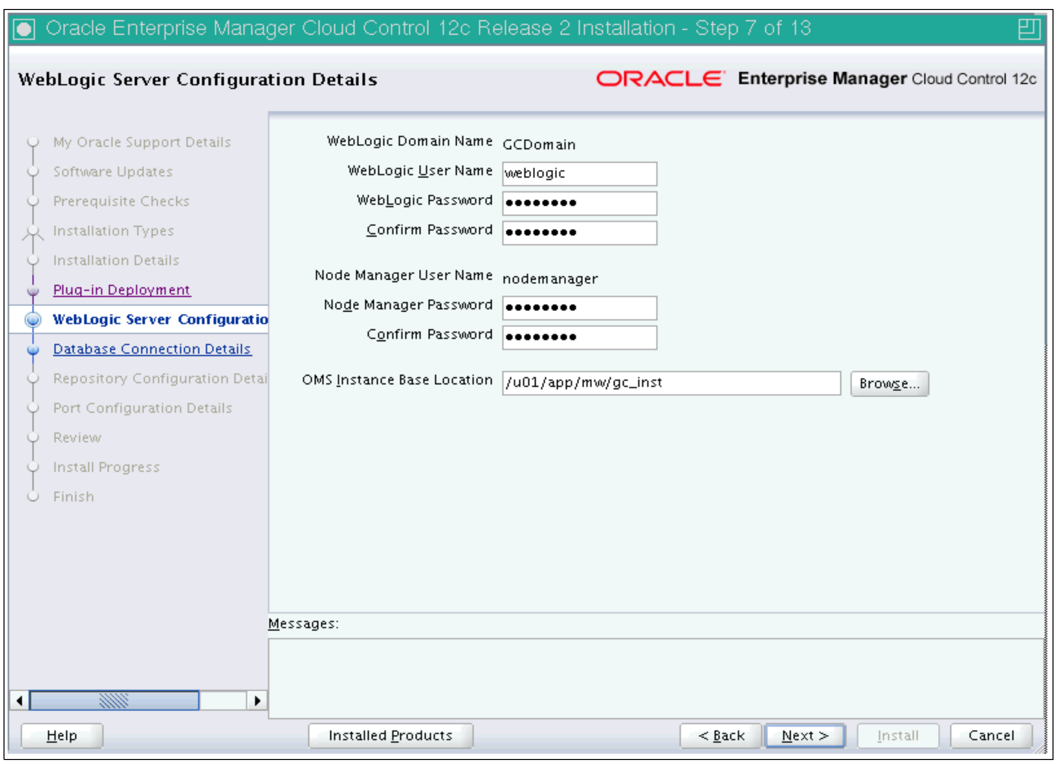


Figure 5-7 WebLogic Server Configuration Details window

As shown in Figure 5-8, we specified the required information for the installed Database connection in the server. We also choose the deployment size as **SMALL**.

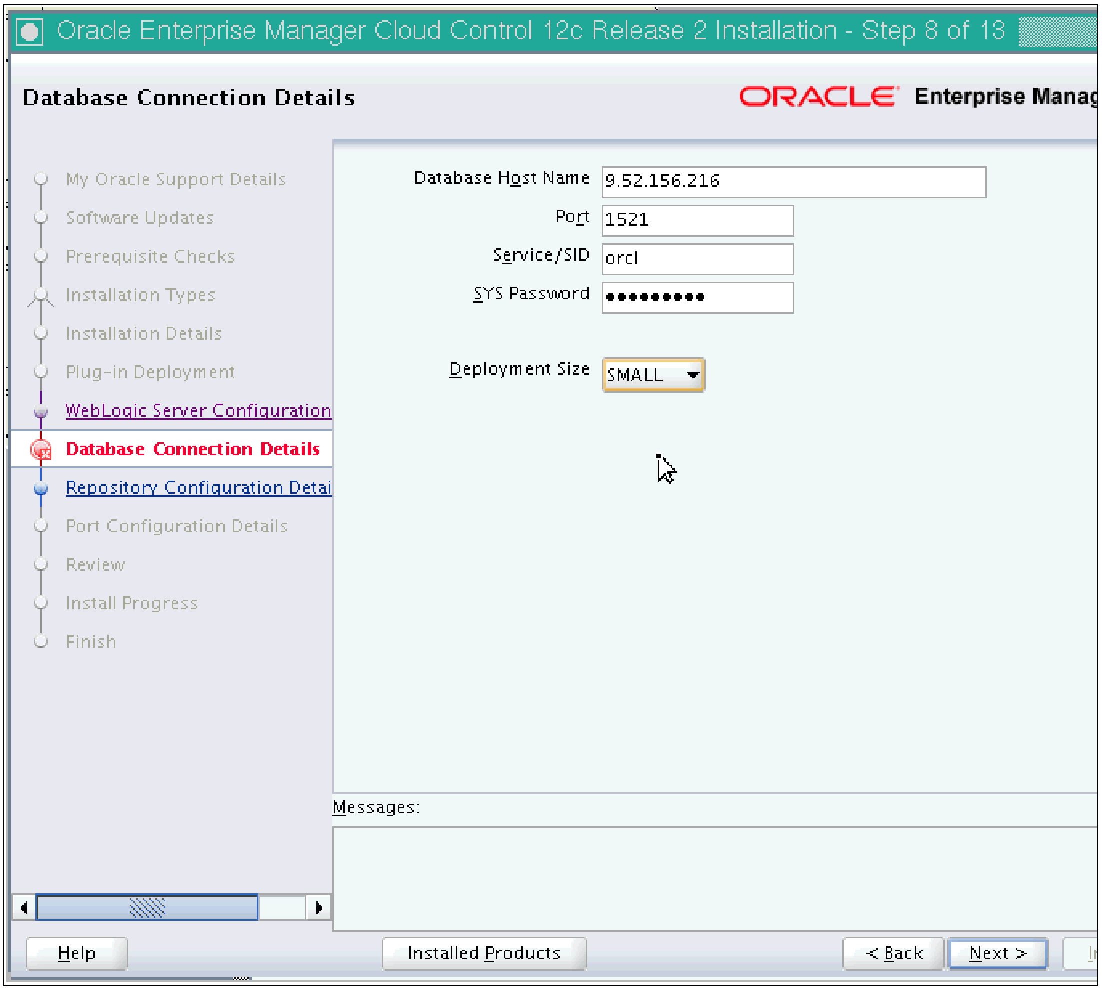


Figure 5-8 Database Connection Details window

When we clicked **Next**, we encountered the error that is shown in Figure 5-9. Although we did not configure the database for Enterprise Manager when we created the database, we still received the error.

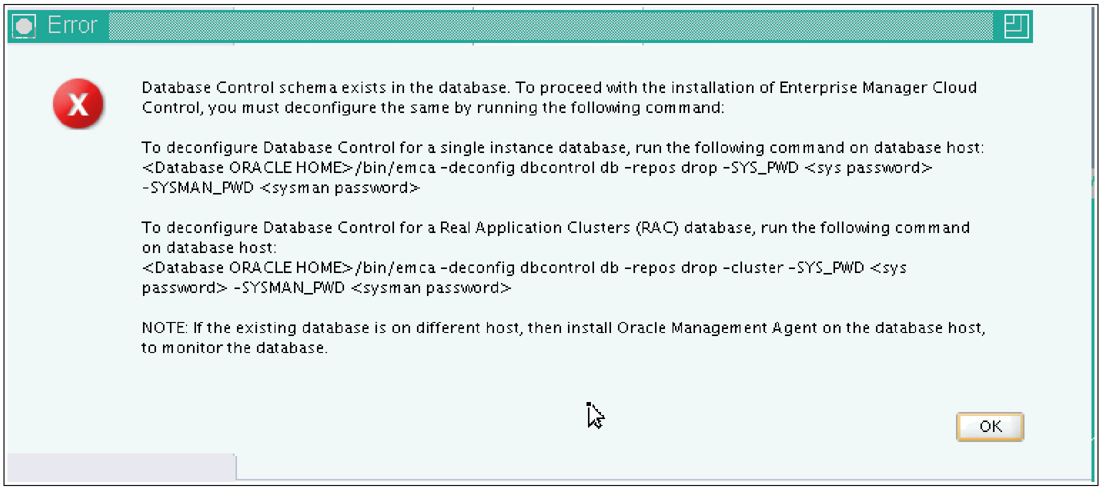
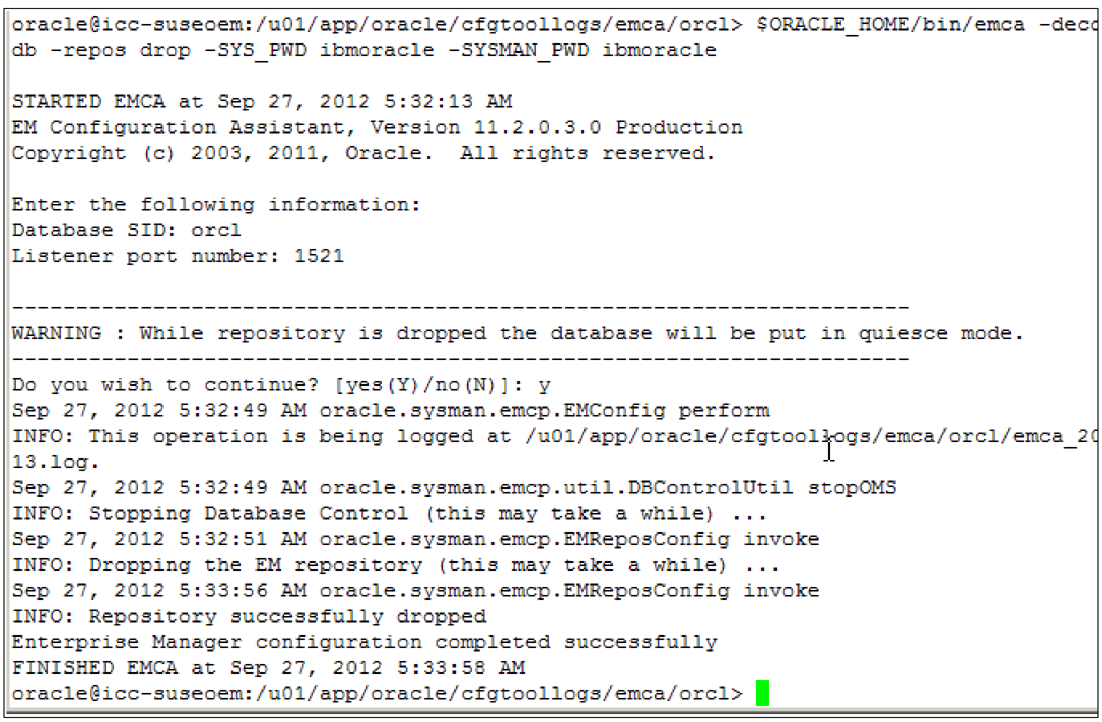


Figure 5-9 Repository existence error

We dropped the repository by using the following command:

```
$ emca -deconfig dbcontrol db -repos drop
```

Figure 5-10 shows that the confirmation that repository was dropped.



```
oracle@icc-suseoem:/u01/app/oracle/cfgtoollogs/emca/orcl> $ORACLE_HOME/bin/emca -dec
db -repos drop -SYS_PWD ibmoracle -SYSMAN_PWD ibmoracle

STARTED EMCA at Sep 27, 2012 5:32:13 AM
EM Configuration Assistant, Version 11.2.0.3.0 Production
Copyright (c) 2003, 2011, Oracle.  All rights reserved.

Enter the following information:
Database SID: orcl
Listener port number: 1521

----------------------------------------------------------------------
WARNING : While repository is dropped the database will be put in quiesce mode.
----------------------------------------------------------------------
Do you wish to continue? [yes(Y)/no(N)]: y
Sep 27, 2012 5:32:49 AM oracle.sysman.emcp.EMConfig perform
INFO: This operation is being logged at /u01/app/oracle/cfgtoollogs/emca/orcl/emca_20
13.log.
Sep 27, 2012 5:32:49 AM oracle.sysman.emcp.util.DBControlUtil stopOMS
INFO: Stopping Database Control (this may take a while) ...
Sep 27, 2012 5:32:51 AM oracle.sysman.emcp.EMReposConfig invoke
INFO: Dropping the EM repository (this may take a while) ...
Sep 27, 2012 5:33:56 AM oracle.sysman.emcp.EMReposConfig invoke
INFO: Repository successfully dropped
Enterprise Manager configuration completed successfully
FINISHED EMCA at Sep 27, 2012 5:33:58 AM
oracle@icc-suseoem:/u01/app/oracle/cfgtoollogs/emca/orcl>
```

Figure 5-10 Repository dropped by using the emca command

The Oracle Cost-Based Optimizer statistics (CBO) gathering job prerequisite appeared, as shown in Figure 5-11 and we choose **Yes** to fix the issue automatically.

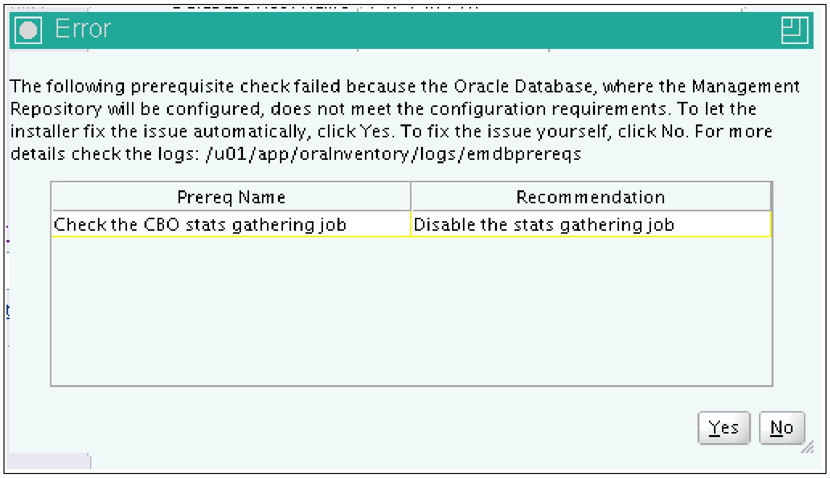


Figure 5-11 CBO statistics gathering error

The Database configuration prerequisite warnings were shown (see Figure 5-12) and we choose to fix the database configurations as recommended by the installation wizard. We used the SQL*PLUS tool to change the parameters and then clicked **OK**.

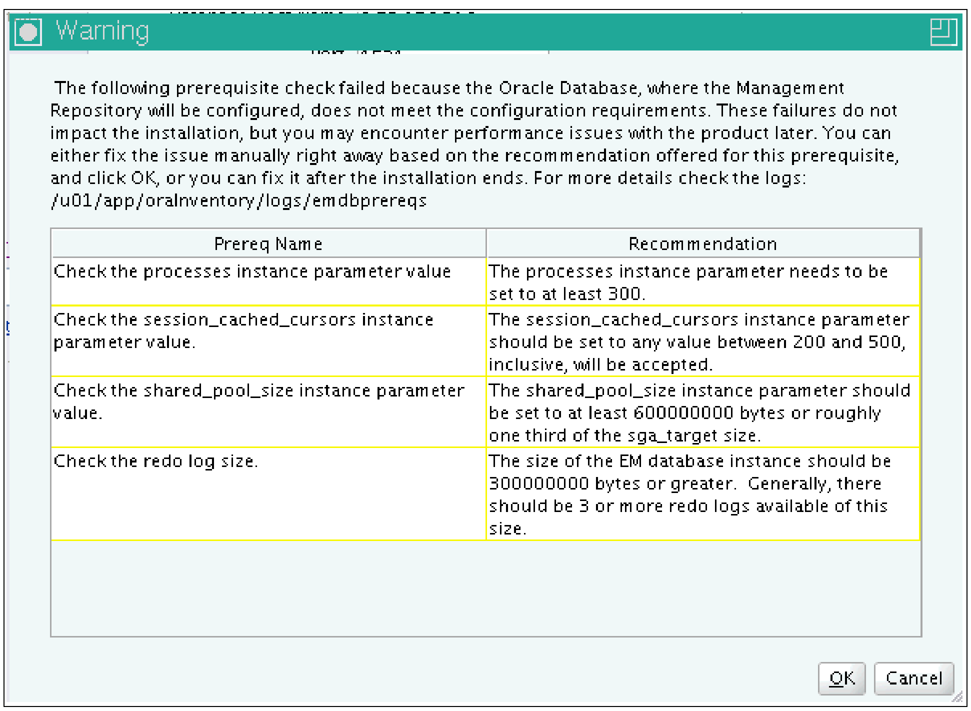


Figure 5-12 Database configuration prerequisite warnings

The Repository configuration password details were entered, as shown in Figure 5-13.

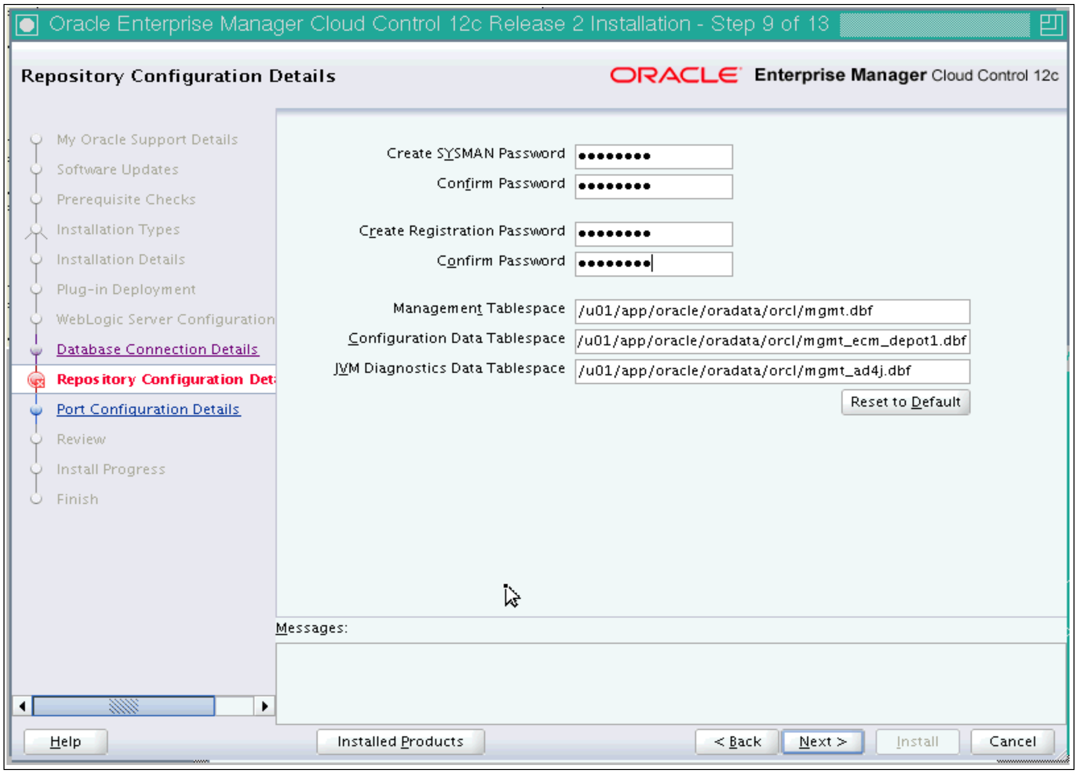


Figure 5-13 Repository Configuration Details window

We accepted the default port values, as shown in Figure 5-14.

Figure 5-14 Port Configuration Details

The configuration values were reviewed and we clicked **Install**, as shown in Figure 5-15.

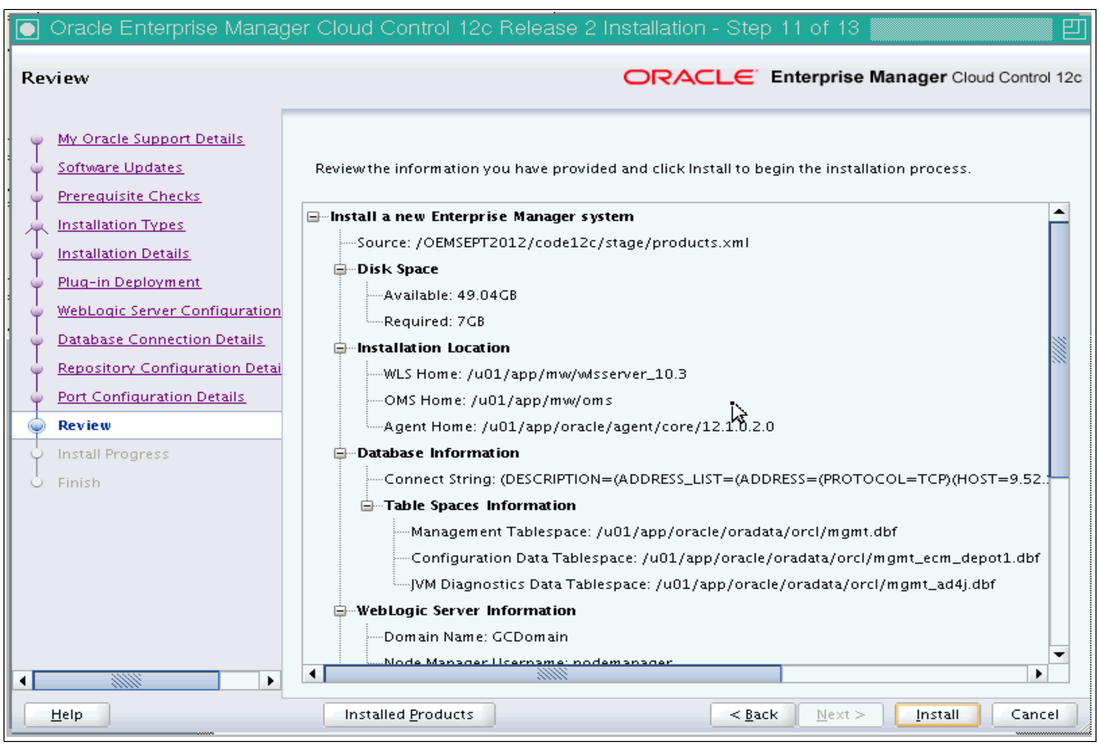


Figure 5-15 Review window

The installation started and we see the progress of the installation, as shown in Figure 5-16.

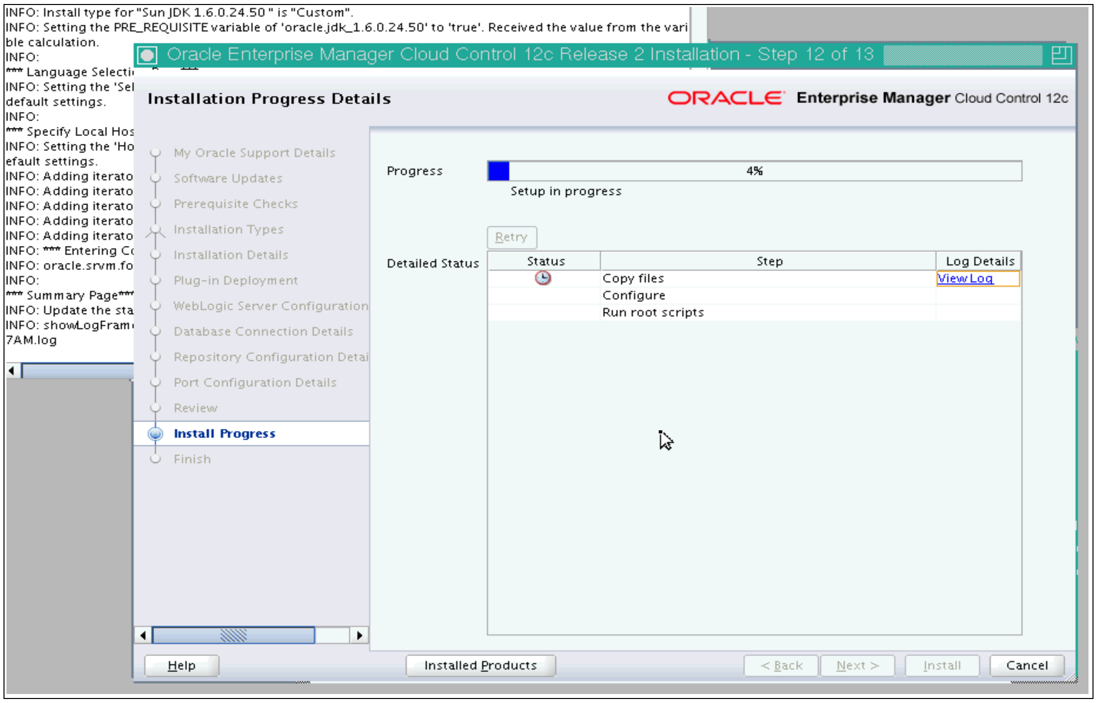


Figure 5-16 Installation Progress Details window

When the installation and configuration completed, we ran the **allroot.sh** command as the root user based on the instructions, as shown in Figure 5-17.

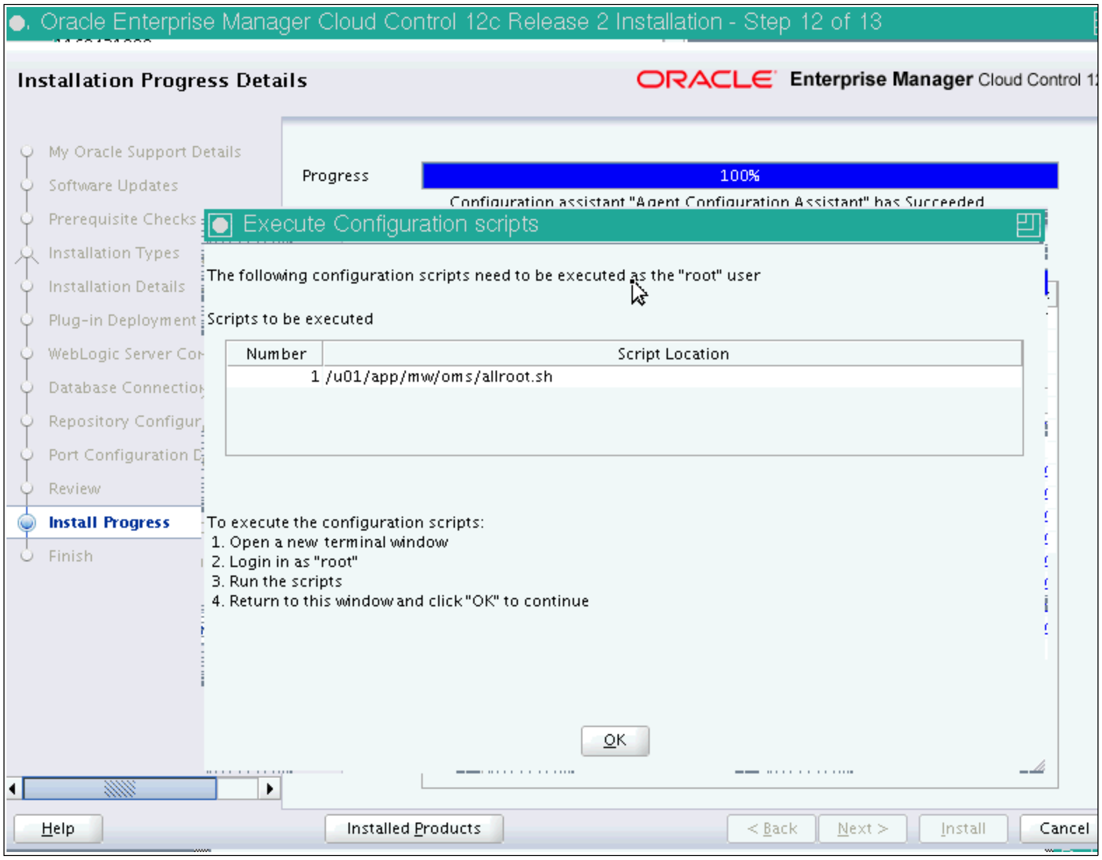


Figure 5-17 Installation Progress Details

The Enterprise Manager Cloud Control configuration and installation took more than 40 minutes to complete and displayed the status. As shown in Figure 5-18, the URL and port number access information for Enterprise Manager Cloud Control and Admin Server was shown.

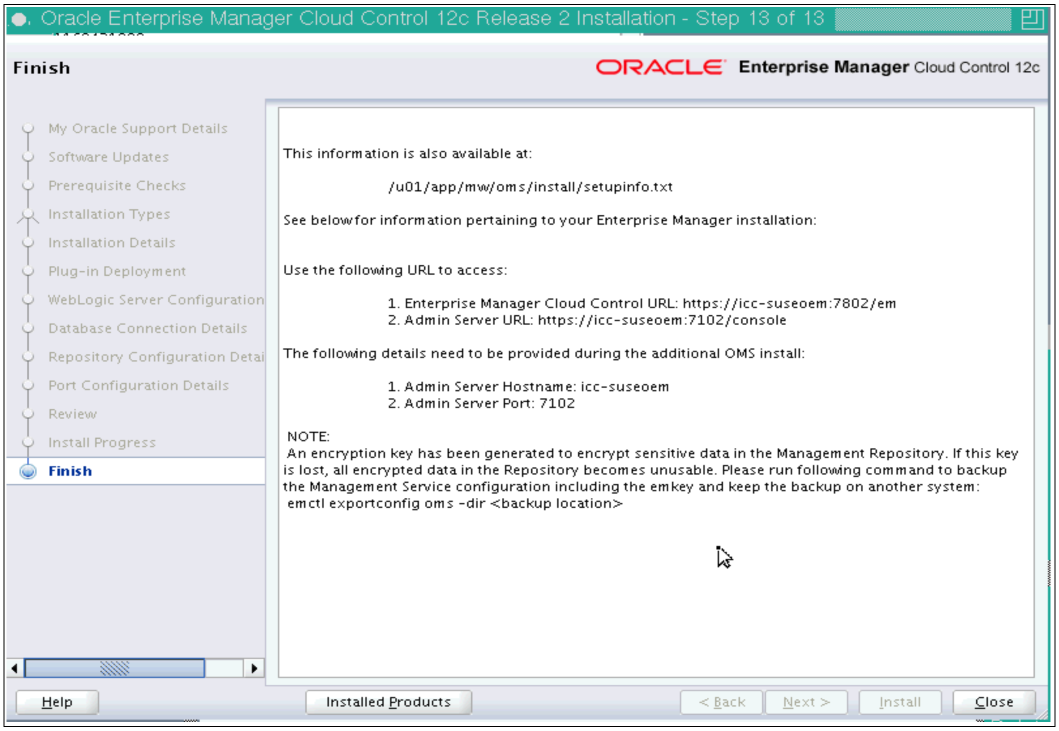


Figure 5-18 Finish window

When we used the Enterprise Manager Cloud Control URL, we were asked to trust and certify and add the exception. Then, the Enterprise Manager Cloud Control window opened and we logged on as `sysman` with the assigned password.

We accepted the license requirement and this completed our Enterprise Manager Cloud Control 12c Server installation.

These tasks are for a simple configuration installation process. Enterprise Manager Cloud Control offers multiple configurations. For more information about advanced installation and configuration options, see the Oracle manuals.

5.3 Updating the Cloud Control Software Library in online mode

Enterprise Manager Cloud Control 12c by default has the agent software installed for the operating system where the 12c is installed. In our example, the agent for the Linux x86-64 is available when the Enterprise Manager Cloud Control 12c is installed on the Linux x86-64 Server. But, the agent to monitor the Oracle databases on Linux on System z architecture is not available in the base installation.

Management Agent software for Linux on System z architecture must be downloaded and stored in the Software Library by using the Self Update console. The Self Update feature is a dashboard that is used to obtain information about new updates and a process flow to review, download, and apply those updates. An Oracle Software Library update can be accomplished in online mode or offline mode and the agent can be deployed through the GUI or in silent installation mode.

The Software Library is a repository that stores software entities such as software patches and application software. The Software Library stores its repository on a file system that is accessible by the Oracle Management Services (OMS). Depending on the usage, the size of this file can reach 50 GB. For more information about advanced features and requirements, see the My Oracle Support (MOS) document *Understanding and Configuring Software Library In 12C Cloud Control* [ID 1368035.1].

5.3.1 Upgrading Software Library by using the Self Update Feature in online

In the following example, we show how the Software Library can be updated in online mode on an Enterprise Manager Cloud Control that is running on x86-64 Linux architecture to get the management agent for Linux on System z.

In general, the process includes the following steps:

1. Set up My Oracle Support Credentials in 12c Cloud Control.
2. Configure the Software Library storage space.
3. Download and apply software updates.
4. Download and apply Management Agent software.

Setting up My Oracle Support Credentials in 12c Cloud Control

Complete the following steps to set up the credentials:

1. Log on to Enterprise Manager Cloud Control 12c
2. Select the Setup menu that is at the upper right of the page. Select **My Oracle Support** → **Set Credentials**.
3. In the My Oracle Support window, enter the User Name and Password fields and click **Apply**.

The My Oracle Support Preferred Credentials are now set up in the 12c Cloud Control.

Configuring the Software Library storage space

The storage location in Software Library represents a repository of files. These files are uploaded by Software Library and we must add at least one upload file storage location. This location can be an OMS Shared location or an OMS Agent location. In our example, we used an OMS shared location and we complete the following steps to set up the location:

1. Identified the `/oracle/emlib` for the Software Library storage space.

2. Logged on to Enterprise Manager Cloud Control 12c.
3. As shown in Figure 5-19, we selected the Setup menu at the upper right of the page. We then selected **Provisioning and Patching** → **Software Library**.

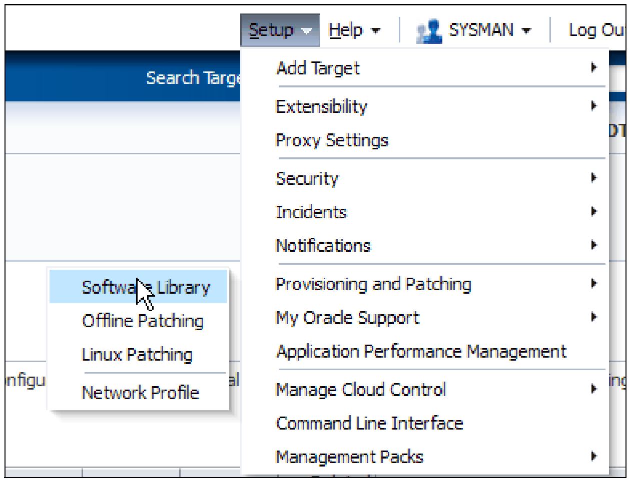


Figure 5-19 Setup option

4. As shown in Figure 5-20, we then selected **Actions** → **Administration**.

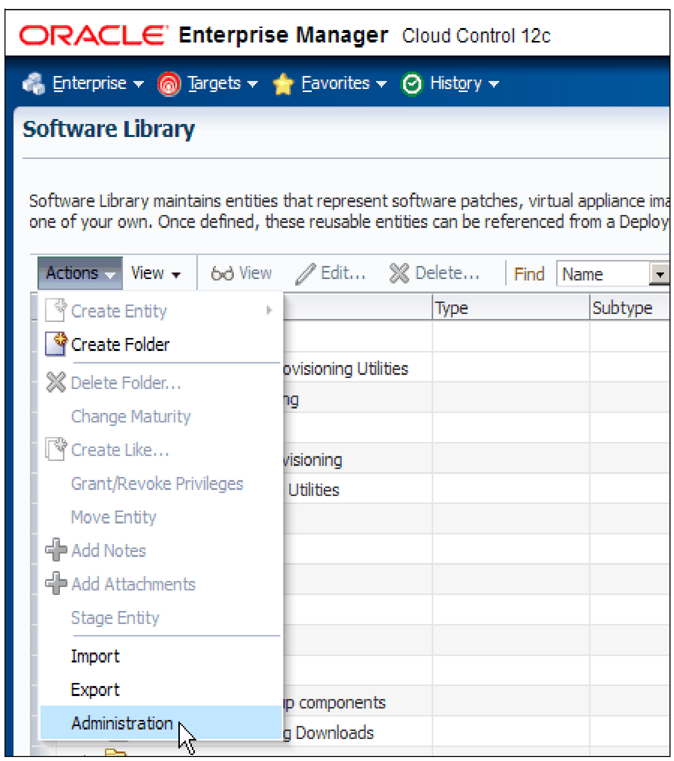


Figure 5-20 Action options

5. As shown in Fig 5-24, in the Administration page, we choose **OMS Shared Filesystem** and selected **Add** to add a new OMS Shared file system.

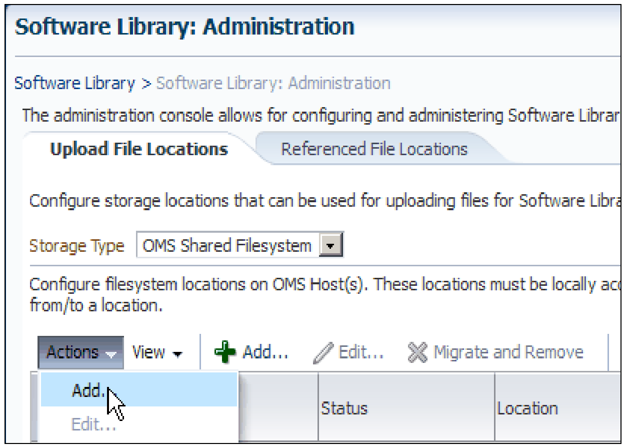


Figure 5-21 Administration options

6. As shown in Figure 5-22, in the Add OMS Shared File System Location panel, we entered the Name (`emlib`) and Location (`/oracle/emlib`) of the OMS host.

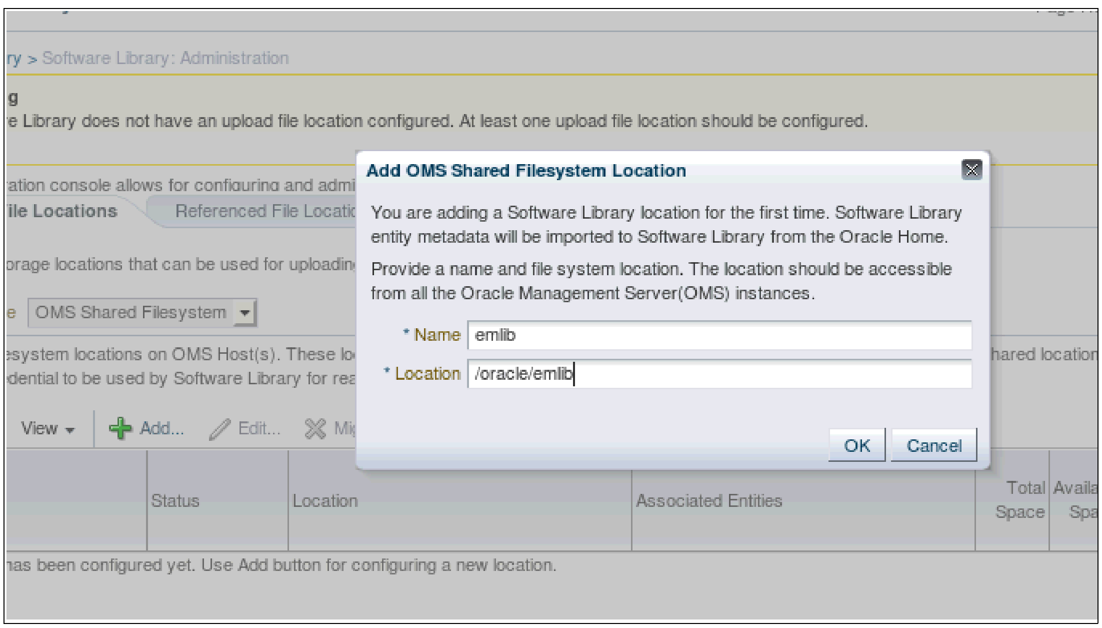


Figure 5-22 Shared Filesystems Location panel

7. When we clicked **OK**, a metadata registration job is submitted (as shown in Figure 5-23) if it is the first time that the process is done. The metadata registration job imports all of the metadata information of all the installed plug-ins from the Oracle home of the OMS.

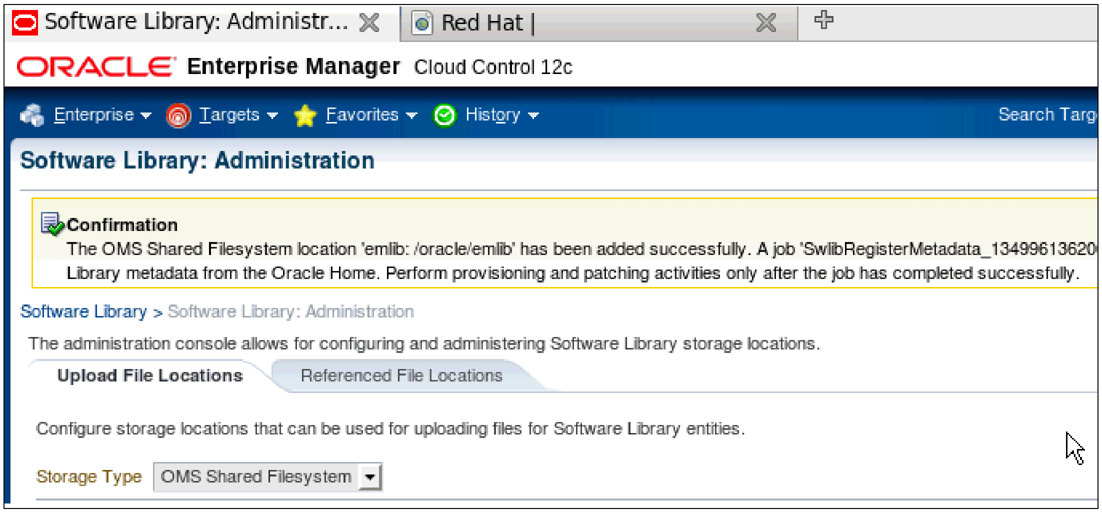


Figure 5-23 Metadata registration job submission

The progress of the job can be tracked from Enterprise menu by selecting **Job**, and then clicking **Activity**.

On the Job Activity page in the Advanced Search region, enter the name of the job, choose **Targetless** as the Target Type, and then click **Search**. Typically, the name of the job starts with `SWLIBREGISTERMETADATA_*`. If this job status shows as succeeded, it implies the software library is configured properly.

Acquiring software updates online

Complete the following steps to acquire software updates online:

1. Ensure that the Cloud Control is set to the online mode, as seen on Figure 5-24 on page 93.
2. Select **Setup** from the menu at the upper right of the page.
   a. Choose **Provisioning and Patching** → **Offline Patching**, then change the setting for Connection to Online.
   b. Select **Software Library**.

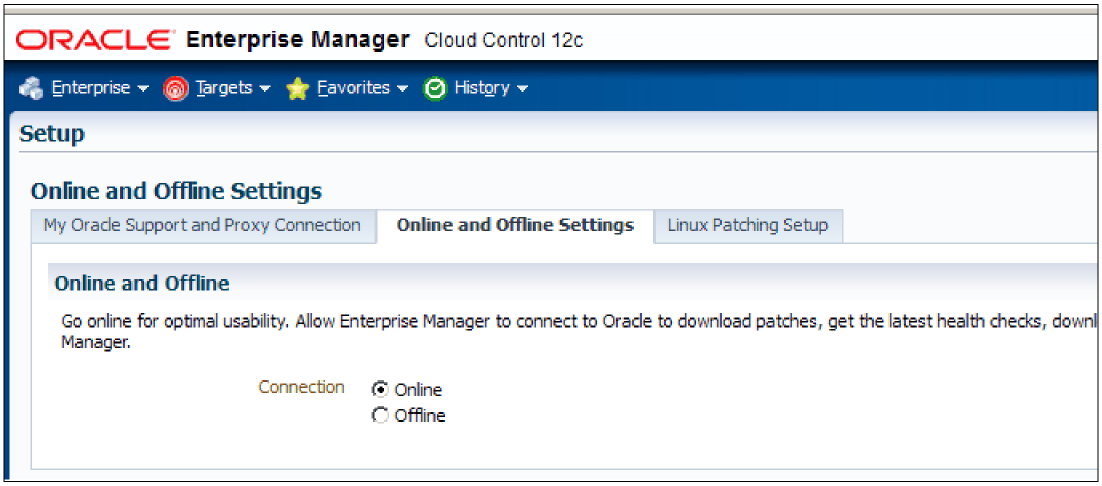


Figure 5-24 Online settings

3. As shown in Figure 5-25, Select the **Setup** menu at the upper right of the page.

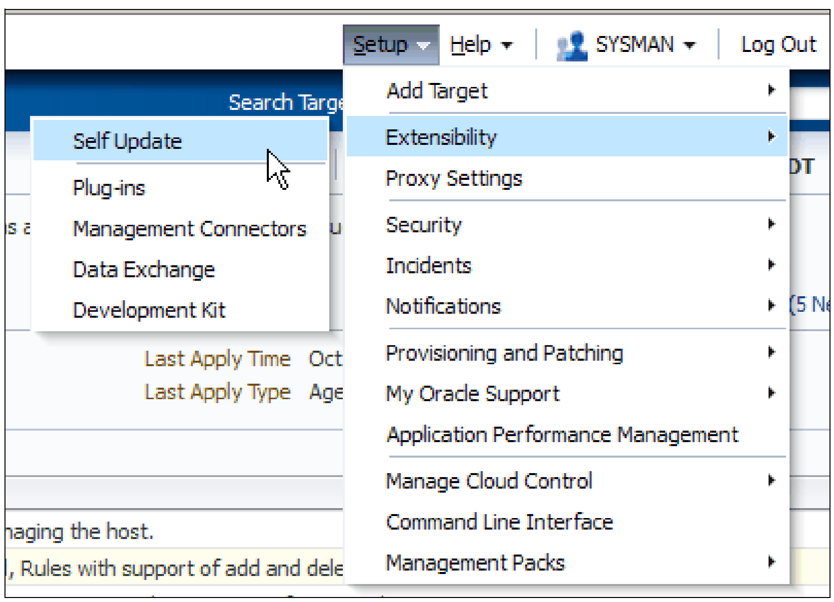


Figure 5-25 Self Update Panel

4. Select **Extensibility** → **Self Update**, and then select **Check Updates** to see the complete list of available updates for the Agent software.

   A background job is submitted to get the new updates from Oracle. The output log shows the output of the job.

Acquiring Management Agent software in online mode

After the list of available updates are populated in the system, we can download the required Management Agent software and then apply that software to the OMS host.

Complete the following steps to download Management Agent software in online mode:

1. As shown in Figure 5-26, Select the **Setup** menu at the upper right of the page and then click **Extensibility** → **Self Update**.

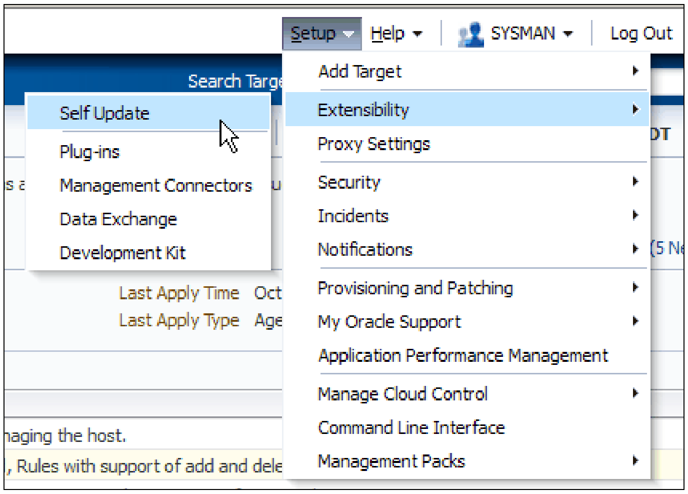


Figure 5-26 Setup menu

2. Select the entity type Agent Software and choose **Open** from the Action menu. The entity type page shows agent software for different platforms.

   a. We selected **IBM: Linux on System z** OS Platform and 12.1.0.2 version from the list of available updates.

   b. We clicked **Download** and scheduled the download job for an immediate run, as shown in Figure 5-27.

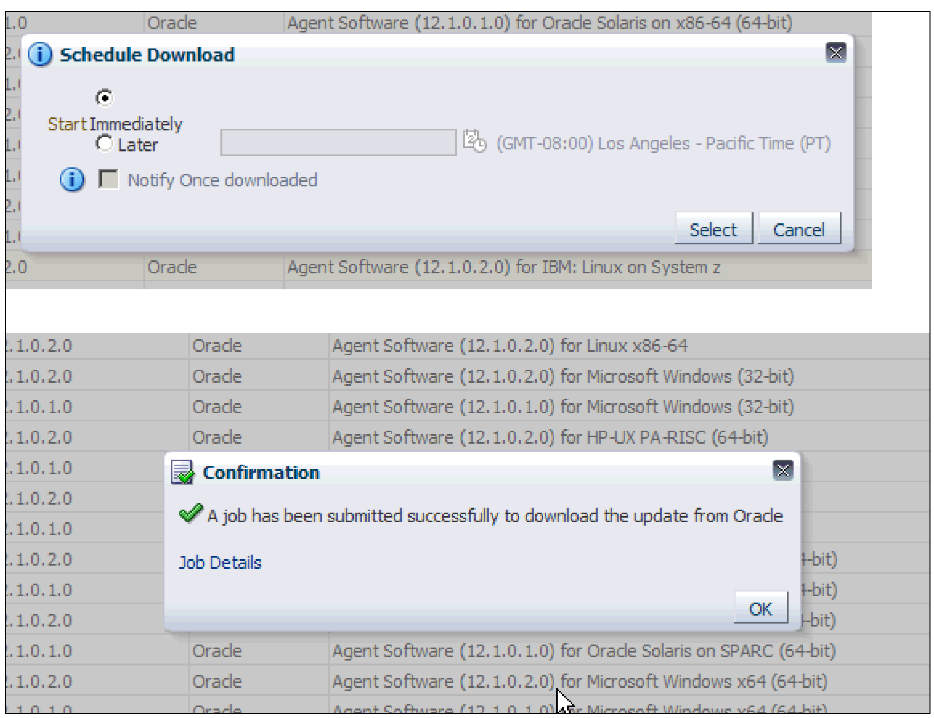


Figure 5-27 Schedule the download

After the successful download, the status is shown as Downloaded.

c. We then selected **Apply** for the Downloaded Agent.

The next step stages the agent software in the Software Library and makes it available to the Add Targets wizard, which we used to install the agent on System z Linux host machines.

After the job is complete, the new status is shown as Applied.

5.4 Updating the Cloud Control Software Library in offline mode

Previously in this chapter, we showed the steps to update the library in online mode. In this section, we describe the process that is used to update the library in the offline mode.

5.4.1 Upgrading Software Library by using the Self Update Feature in offline mode

In the following example, we describe how the Software Library can be updated in offline mode on an Enterprise Manager Cloud Control that is running on x86-64 Linux architecture to get the management agent for Linux on System z. Oracle requires the use of the Enterprise Manager command-line interface (EMCLI to update the EM Cloud Control Software updates.

In general, the process includes the following tasks:

1. Set up the EMCLI.
2. Configure the Software Library storage space.
3. Download and apply software updates offline.
4. Download and apply Management Agent software offline.

Installing and setting up the EMCLI

In the following example, we describe the process that is used to install and set up the EMCLI. By using the EMCLI, we can update the Software Library in offline mode. The EMCLI client can be installed on any machine in the network. In our example, we installed the client on the Enterprise Manager Cloud Control Server. The only requirement was to have Java version 1.6.0_25 or greater.

We completed the following steps to install and set up the EMCLI client:

1. Connect to the EMCLI download package from the Enterprise Manager Cloud Console by using the following URL:

   `https://<omsmachine.domian>:<port>/em/console/emcli/download`

   In the window that opens, click **Download the Enterprise Manager CLI kit to your workstation**.

2. Download the `emclikit.jar` file.

3. Ensure that the Java version is 1.6.0_25 and JAVA_HOME and PATH is set up appropriately to use that Java level.

4. Use the following command (as seen in Figure 5-28 on page 96) to install EMCLI:

   ```
   java -jar emclikit.jar client -install_dir=/zCode/patches
   ```

   Specify `/zCode/patches` as the installation directory.

```
[oracle@zs2120-11 patches]$ ls -alF emc*
-rw-r--r--. 1 oracle oinstall 1416741 Oct 13 13:39 emclikit.jar
[oracle@zs2120-11 patches]$
```

```
[oracle@zs2120-11 patches]$ java -jar emclikit.jar client -install_dir=/zCode/patches
Oracle Enterprise Manager 12c Release 2.
Copyright (c) 1996, 2012 Oracle Corporation.  All rights reserved.

EM CLI client-side install completed successfully.
```

Figure 5-28 Command to install the EMCLI client

5. Set up the EMCLI client by running the following command from where the EMCLI client is installed:

   ./emcli setup -url="https://*9.12.5.131:7802*/em" -username=sysman -dir=/*zCode/patches* -trustall -certans=yes

6. Use the **./emcli setup** command to see how the EMCLI client was set up in that environment, as shown in Figure 5-29.

```
Emcli setup successful
[oracle@zs2120-11 patches]$ ./emcli setup
Oracle Enterprise Manager Cloud Control 12c Release 2.
Copyright (c) 1996, 2012 Oracle Corporation and/or its affiliates. All rights reserve

Instance Home          : /zCode/patches/.emcli
Verb Jars Home         : /zCode/patches/.emcli
EM URL                 : https://9.12.5.131:7802/em
EM user                : sysman
Trust all certificates : true
Auto login             : false
```

Figure 5-29 Completion of the EMCLI installation

Configuring the Software Library storage space

The storage location in Software Library represents a repository of files. These files are uploaded by Software Library and we must add at least one upload file storage location. This location can be an OMS Shared location or an OMS Agent location. In our case, we decided to use an OMS shared location. We used the process that is described in "Configuring the Software Library storage space" on page 89 to set up the storage space as `/zCode/EMGridSWLib`, as shown in Figure 5-30.

```
[oracle@zs2120-11 EMGridSWLib]$ pwd
/zCode/EMGridSWLib
[oracle@zs2120-11 EMGridSWLib]$ ls
0   11  14  17  2   22  25  28  30  33  36  39  41  44  47  5   52  55  58  60  63
1   12  15  18  20  23  26  29  31  34  37  4   42  45  48  50  53  56  59  61  64
10  13  16  19  21  24  27  3   32  35  38  40  43  46  49  51  54  57  6   62  65
[oracle@zs2120-11 EMGridSWLib]$
```

Figure 5-30 Content of the library

Acquiring software updates offline

Complete the following steps to update the software in Offline mode:

1. Ensure that the Cloud Control Connection setting is set to the offline mode.
2. Select **Setup** at the upper right of the page. Choose **Provisioning and Patching** → **Offline Patching** and change the setting for Connection to Offline.
3. Select **Setup** at the upper right of the page.
4. Choose **Extensibility** → **Self Update**, and then click **Check Updates** to get the complete list of available updates for the Agent software. It is indicated in the window that the connection status is Offline.

   As shown in Figure 5-31, a link is provided in a panel to download the latest Oracle updates catalog. Instructions also are included about how to import the catalog updates into Enterprise Manager.

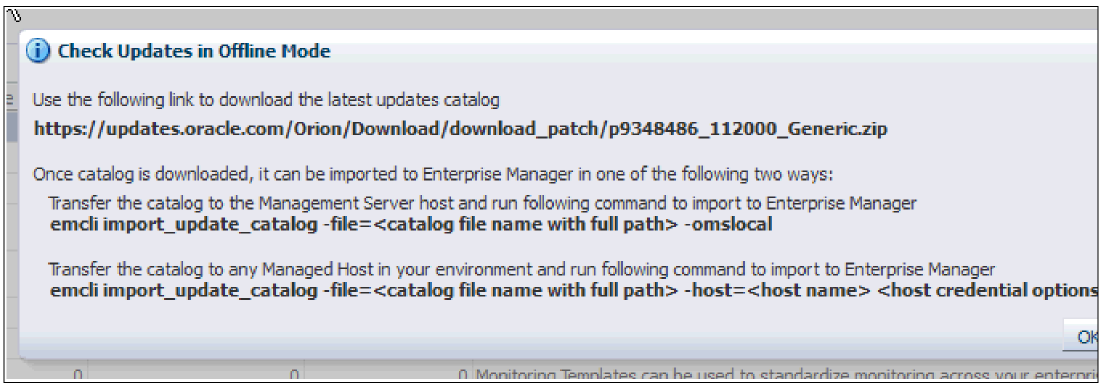


Figure 5-31 Link to the latest updates

5. By using the provided link (you must provide MOS logon credentials), the `.zip` file is downloaded, as shown in Figure 5-32.

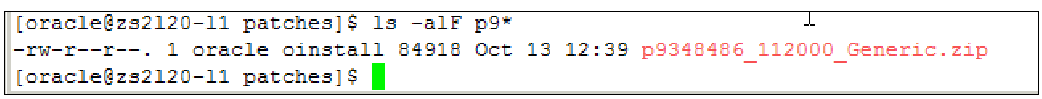


Figure 5-32 Downloaded compressed file

6. Use the **`emcli import_update_catalog`** command to specify the location for the downloaded patch file and `omslocal` options, as shown in Figure 5-33.

```
[oracle@zs2120-11 patches]$ ./emcli import_update_catalog -file="/zCode/patches/p9348
ric.zip" -omslocal
Processing catalog for Provisioning Bundle
Processing catalog for Agent Software
Processing update: Agent Software - Agent Software (12.1.0.2.0) for Microsoft Windows
Processing update: Agent Software - Agent Software (12.1.0.1.0) for Microsoft Windows
Processing update: Agent Software - Agent Software (12.1.0.2.0) for HP-UX PA-RISC (64
Processing update: Agent Software - Agent Software (12.1.0.1.0) for HP-UX PA-RISC (64
Processing update: Agent Software - Agent Software (12.1.0.2.0) for Linux x86
Processing update: Agent Software - Agent Software (12.1.0.1.0) for Linux x86
Processing update: Agent Software - Agent Software (12.1.0.2.0) for Oracle Solaris on
)
Processing update: Agent Software - Agent Software (12.1.0.1.0) for Oracle Solaris on
)
Processing update: Agent Software - Agent Software (12.1.0.2.0) for Oracle Solaris on
Processing update: Agent Software - Agent Software (12.1.0.1.0) for Oracle Solaris on
Processing update: Agent Software - Agent Software (12.1.0.2.0) for Microsoft Windows
Processing update: Agent Software - Agent Software (12.1.0.1.0) for Microsoft Windows
Processing update: Agent Software - Agent Software (12.1.0.1.0) for Linux x86-64
Processing update: Agent Software - Agent Software (12.1.0.2.0) for IBM AIX on POWER
)
Processing update: Agent Software - Agent Software (12.1.0.1.0) for IBM AIX on POWER
)
Processing update: Agent Software - Agent Software (12.1.0.2.0) for IBM: Linux on Sys
Processing update: Agent Software - Agent Software (12.1.0.1.0) for IBM: Linux on Sys
Processing update: Agent Software - Agent Software (12.1.0.2.0) for HP-UX Itanium
Processing update: Agent Software - Agent Software (12.1.0.1.0) for HP-UX Itanium
Processing catalog for Management Connector
Processing update: Management Connector -
             IBM TEC Connector - 12.1.0.2.0

Processing update: Management Connector -
             Microsoft SCOM R2 Connector - 12.1.0.2.0
```
Figure 5-33 Location of the download patches

Acquiring Management Agent software in offline mode

After the list of available updates are populated in the system, we can download the required Management Agent Software and then apply that to the OMS host.

Complete the following steps to download the software and apply it to the OMS host:

1. Select **Setup** in the upper right of the page and then click **Extensibility** → **Self Update**.
2. Select the entity type **Agent Software** and then choose **Open** from the Action menu. The entity type page shows agent software for different platforms.
3. Select **IBM: Linux on System z** as the OS Platform and 12.1.0.2 version from the list of available updates.

   A panel shows the link to download the selected agent and instructions about how to import the agent software updates into Enterprise Manager. Use the link to download the `.zip` file.
4. Use the **`emcli import_update_catalog`** command to specify the location for the downloaded patch file and `omslocal` options, as shown in the Figure 5-34 on page 99.

```
[oracle@zs2120-l1 patches]$ ls -alF p14570374_112000_Generic.zip
-rw-r--r--. 1 oracle oinstall 285350480 Oct 13 14:52 p14570374_112000_Generic.zip
[oracle@zs2120-l1 patches]$

[oracle@zs2120-l1 patches]$ ./emcli import_update -omslocal -file="/zCode/patches/p
eneric.zip"
Processing update: Agent Software - Agent Software (12.1.0.2.0) for IBM: Linux on S
Operation completed successfully. Update has been uploaded to Enterprise Manager. P
f Update Home to manage this update.
[oracle@zs2120-l1 patches]$
```

Figure 5-34 Update is loaded

In the Self Update window, the status is shown as Downloaded for the IBM: Linux on System z agent type.

5. Select **Apply** for the Downloaded Agent.

   The agent software is staged in the Software Library and is available to the Add Targets wizard, which we used to install the agent on System z Linux host machines.

   After the job is completed, the status is shown as Applied.

5.5 Deploying the Agents from Cloud Control console

Oracle Management Agent 12c for Linux on System z hosts can be deployed from Enterprise Manager Cloud Control 12c console or by silent installation method. In this section, we describe the processes to deploy from Cloud Control 12c console.

Oracle recommends the use of the Add Host Targets wizard, which converts a host that is not managed to a managed host in the Enterprise Manager system by installing an Oracle Management Agent 12c. For Real Application Cluster (RAC) with multiple nodes, Management Agent must be installed on each of the nodes separately in the Add Host Targets wizard by adding each node explicitly as a destination host.

For more information about hardware and software requirements for installing Cloud Control Agents, see the Oracle documentation.

The following minimum requirements are specified by Oracle for a stand-alone Management Agent:

- 2 CPU cores per host
- 512 MB of RAM
- 1 GB of free hard disk space
- 400 MB of `/tmp` directory space at the destination host

The network between Cloud Control Server where the OMS is running and the destination hosts should be accessible.

We used the **`ping`** by host name to make sure that the OMS and hosts can be reached.

We used the following process to install Oracle Management Agent 12c for Linux on System z from Enterprise Manager Cloud Control 12c.

To add or install an Agent on a host, the software distribution of the Agent that corresponds to the host's platform must be available in the Software Library.

Complete the following steps to verify the availability of Linux on System z agents availability in the EM Cloud Control Server:

1. Log on to Enterprise Manager Cloud Control 12c.
2. Select **Setup** → **Extensibility** → **Self Update**, as shown in Figure 5-35.

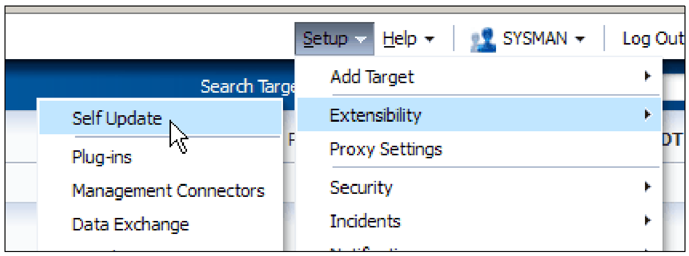


Figure 5-35 Self-update options

3. In the Status section of the Self Update window, select **Agent Software** as the type, as shown in Figure 5-36

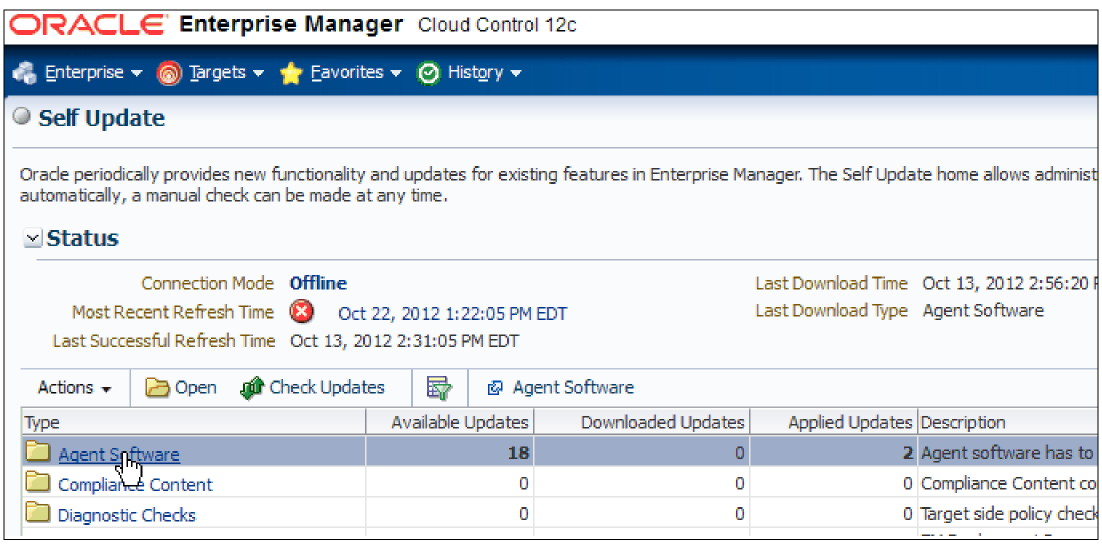


Figure 5-36 Select the agent

4. In the Agent Software Updates section (see Figure 5-37 on page 101), you can see that Agent Software for the Linux on System z shows a status of Applied. When the rows are highlighted, the bottom of the window shows the status, as it did when the agent software was available, downloaded, and applied.

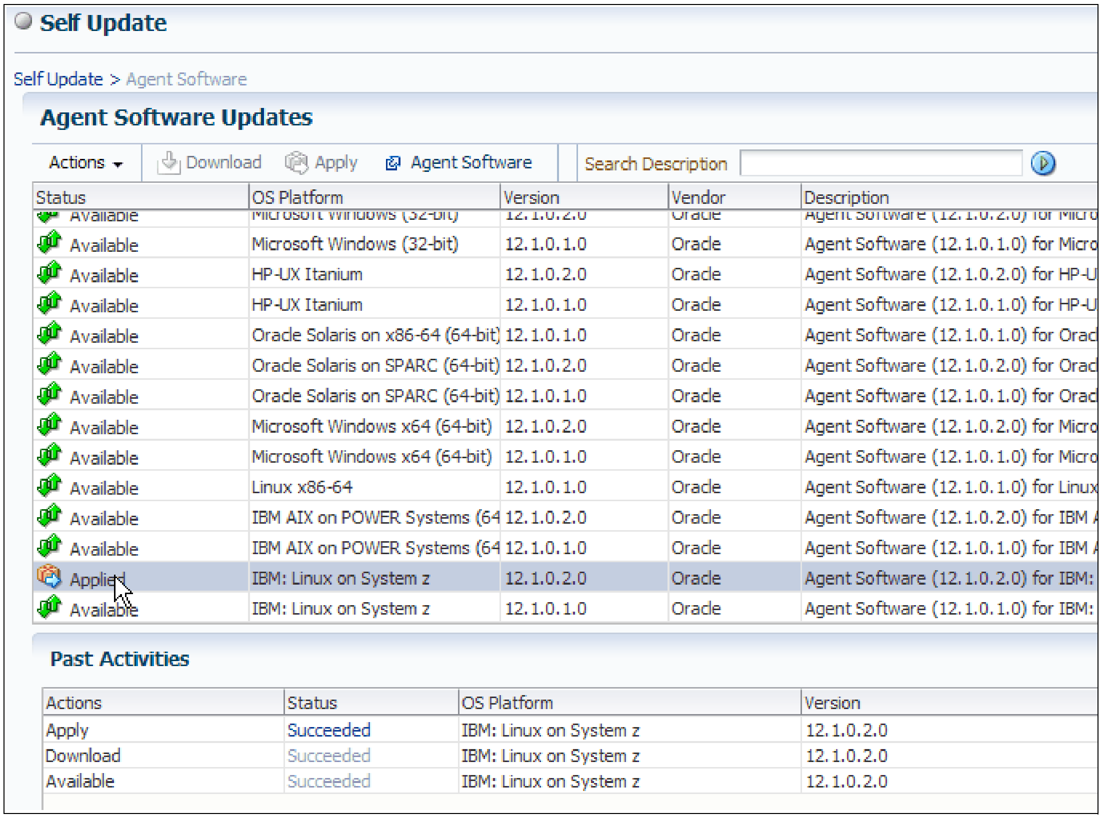


Figure 5-37 Status of the agent

5. To deploy the agent, select **Setup** → **Add Target** → **Add Targets Manually**, as shown in Figure 5-38.

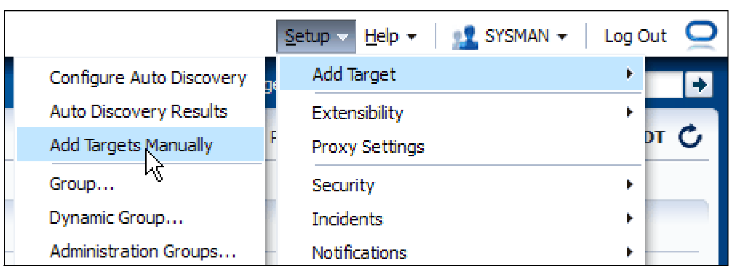


Figure 5-38 Add the target

6. On the Add Targets Manually page, select **Add Host Targets** and then click the **Add Host** button.
7. On the Host and Platform page (see Figure 5-39 on page 102), complete the following steps:
   a. Accept the default name that is assigned for this session.
   b. Click **Add** and enter the fully qualified name of the host. Select **IBM: Linux on System z** as the platform of the host on where we want to install the Management Agent. Select **Next**.

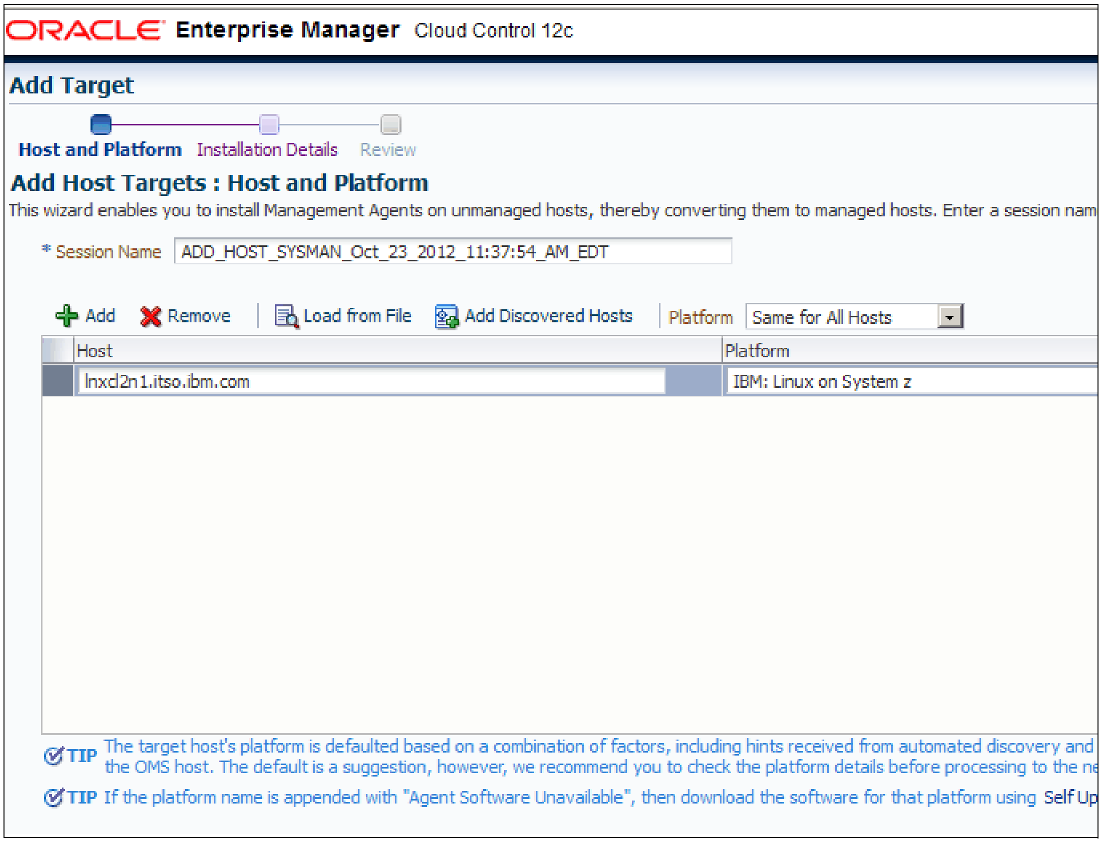


Figure 5-39 Add host

8. On the Installation Details page (see Figure 5-40), complete the following steps:
   a. In the Deployment Type section, select **Fresh Agent Install**.
   b. In the Installation Details section, enter the path to the base directory for Installation Base Directory (the software binaries, security files, and inventory files of Management Agent are copied here). In our case, it is `/u01/oracle/agentHome`.
   c. For the Instance Directory, we accept the default instance directory location (all Management Agent-related configuration files can be stored here). In our case, the `/u01/oracle/agentHome/agent_inst` directory is used.

> **Recommendation:** Oracle recommends that the instance directory is maintained inside the installation base directory.

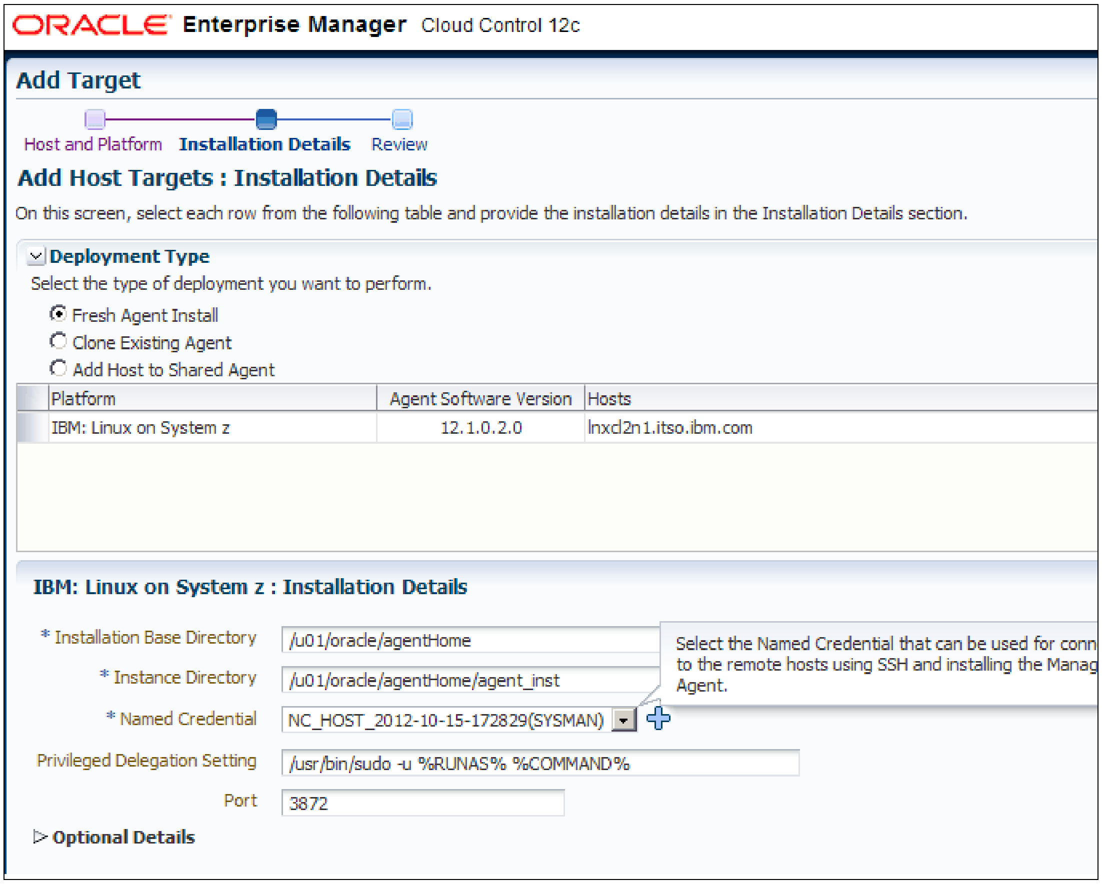


Figure 5-40 Installation details

9. From the Named Credential list, add a new profile of which the credentials are used for setting up the SSH connectivity between the OMS and the remote host. They are also used for installing a Management Agent, as shown in Figure 5-41. Click **Next**.

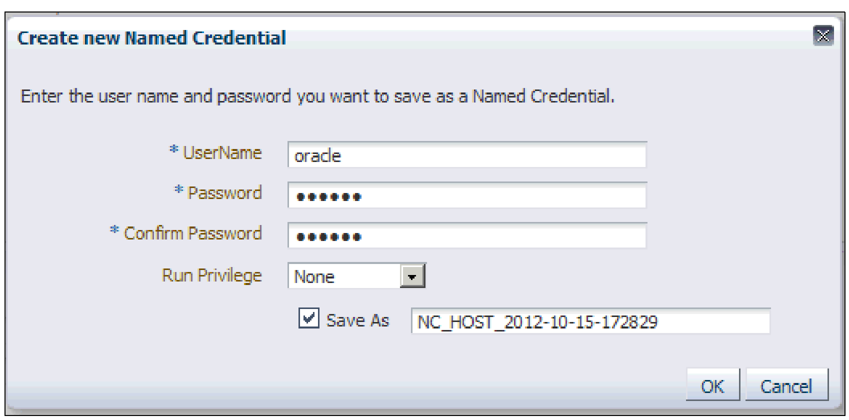


Figure 5-41 UserName

10. On the Review page (see Figure 5-42), review the details and then click **Deploy Agent** to install the Management Agent.

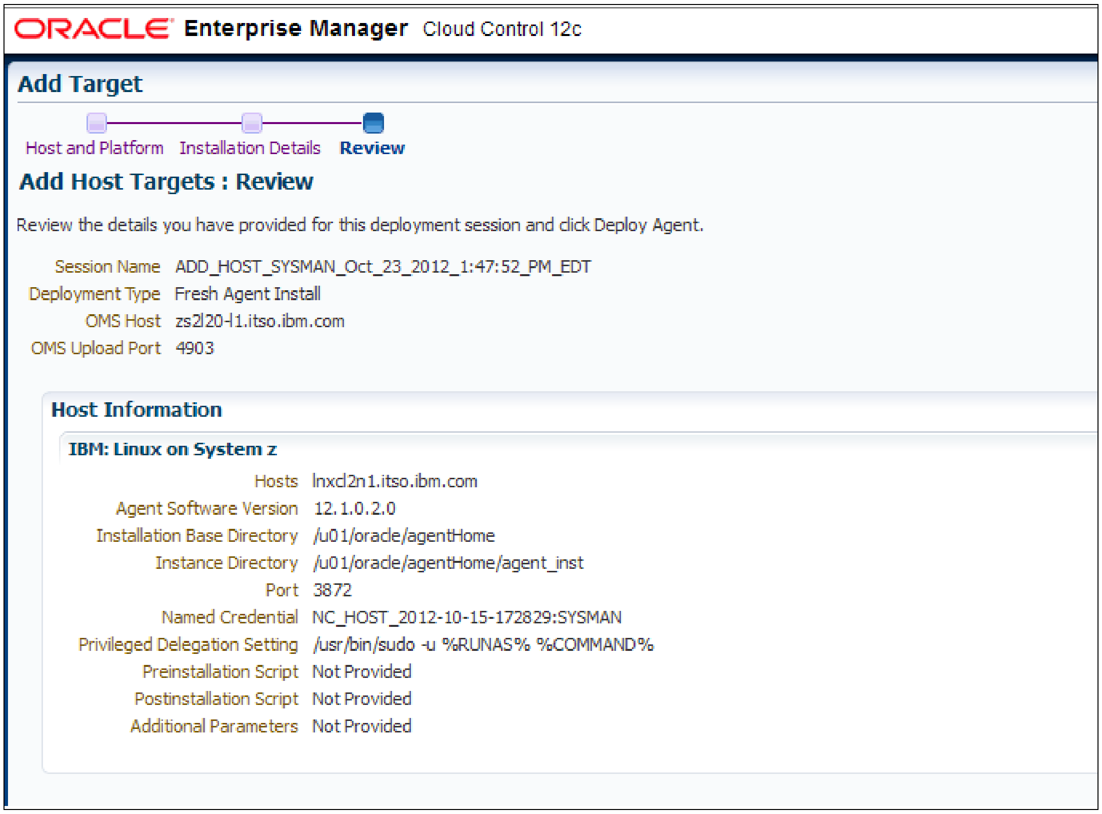


Figure 5-42 Review panel

The progress of installation can be monitored in the Add Hosts Status window, as shown in Figure 5-43.

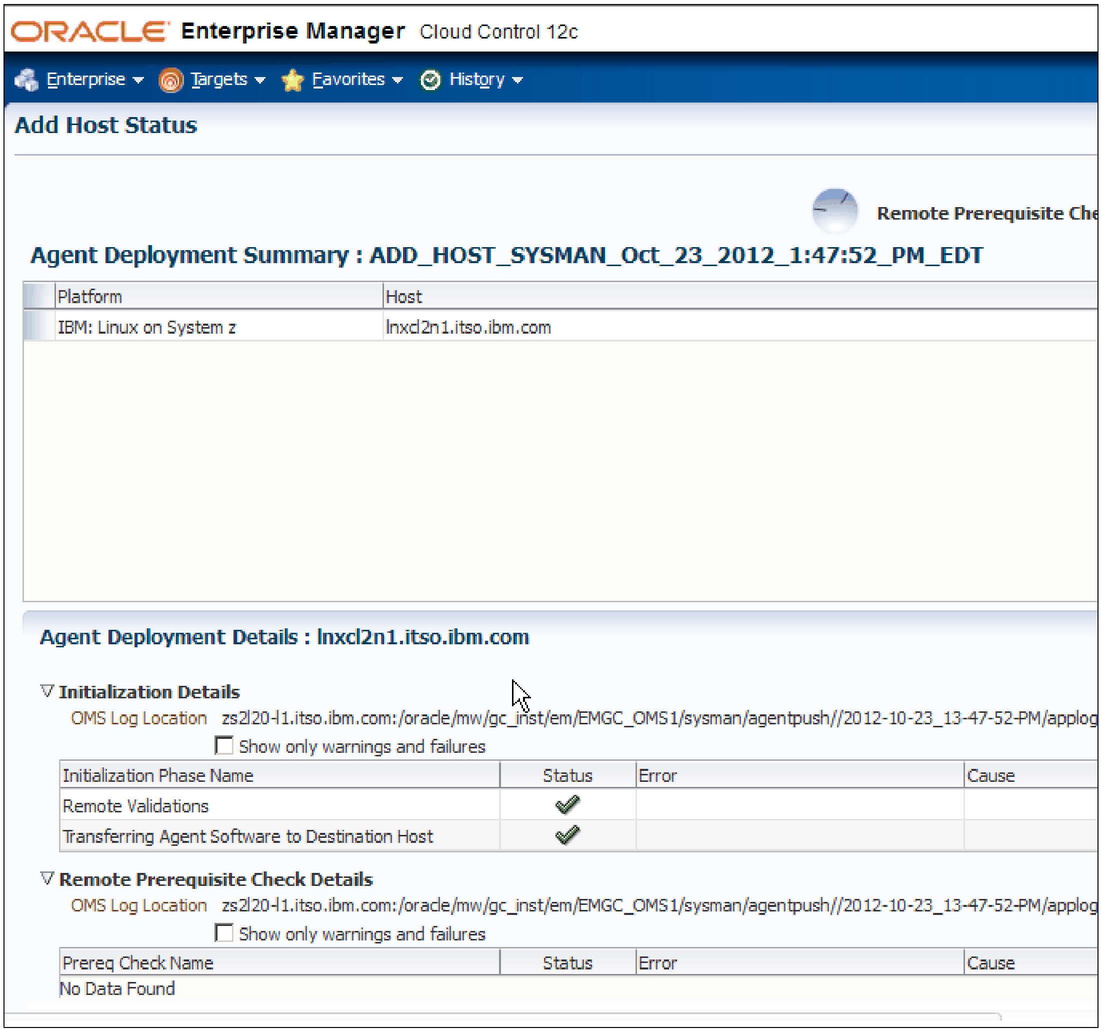


Figure 5-43 Add host status

During the prerequisite check stage, the deployment failed concerning `root.sh` authorization messages, as shown in Figure 5-44.

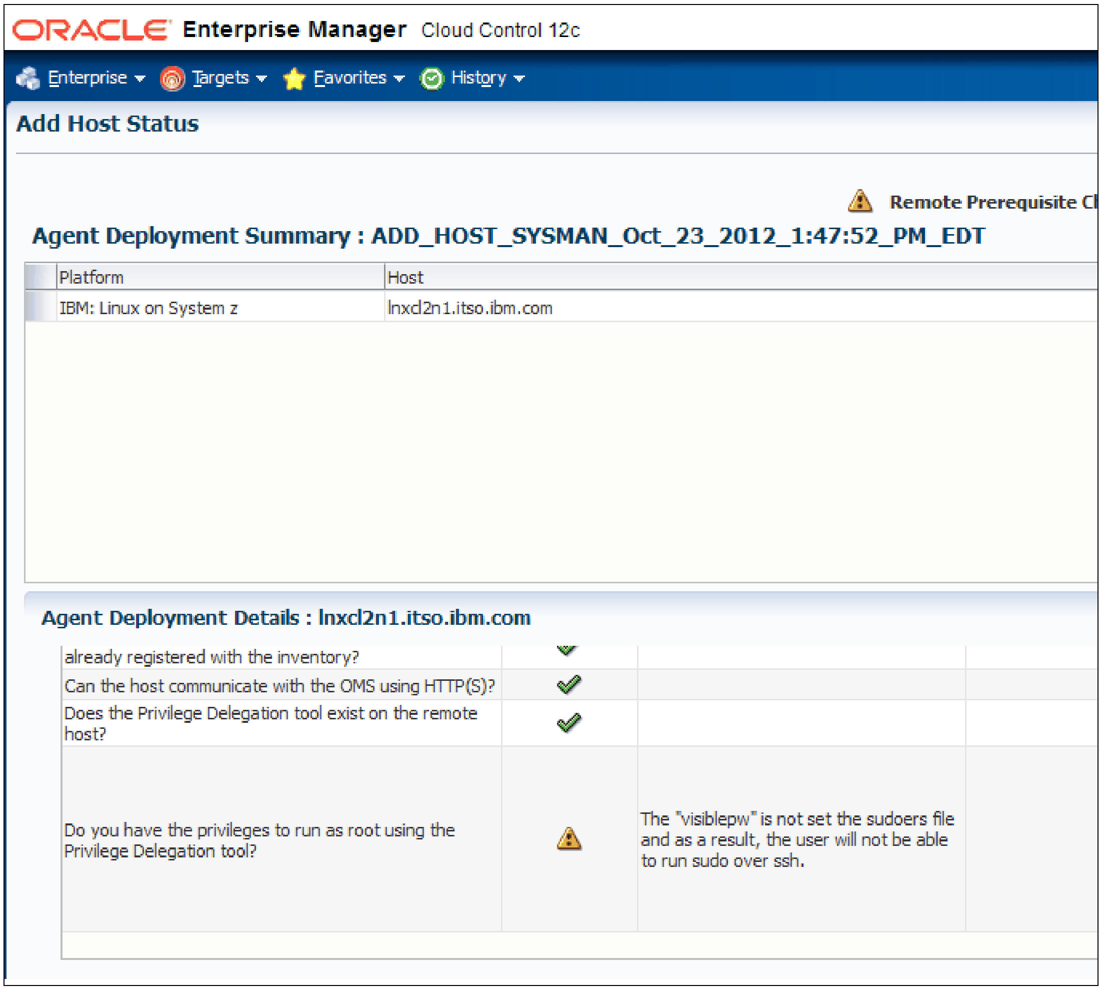


Figure 5-44 Prerequisite check

Continue the installation by selecting the **Continue all hosts** option, as shown in Figure 5-45.

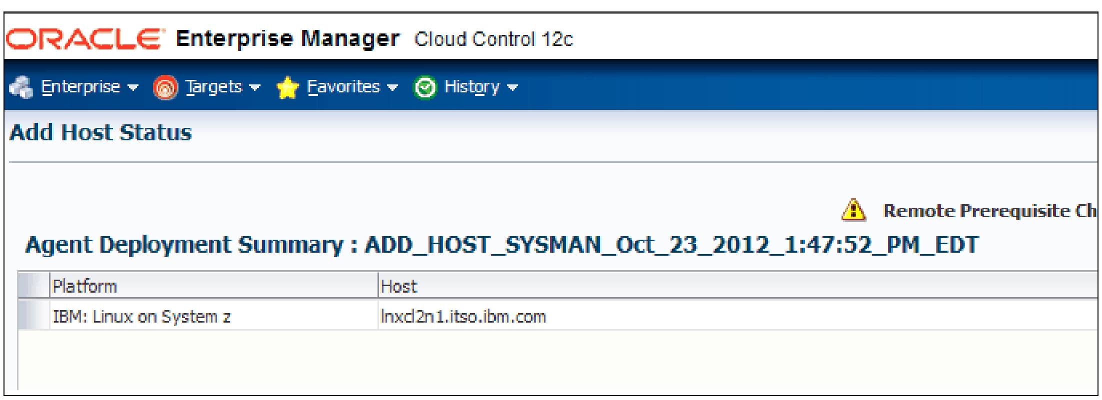


Figure 5-45 Continue the host status option

An Agent Deployment Summary message shows when the process is complete. The Deployment of the agent shows the status that is shown in Figure 5-46.

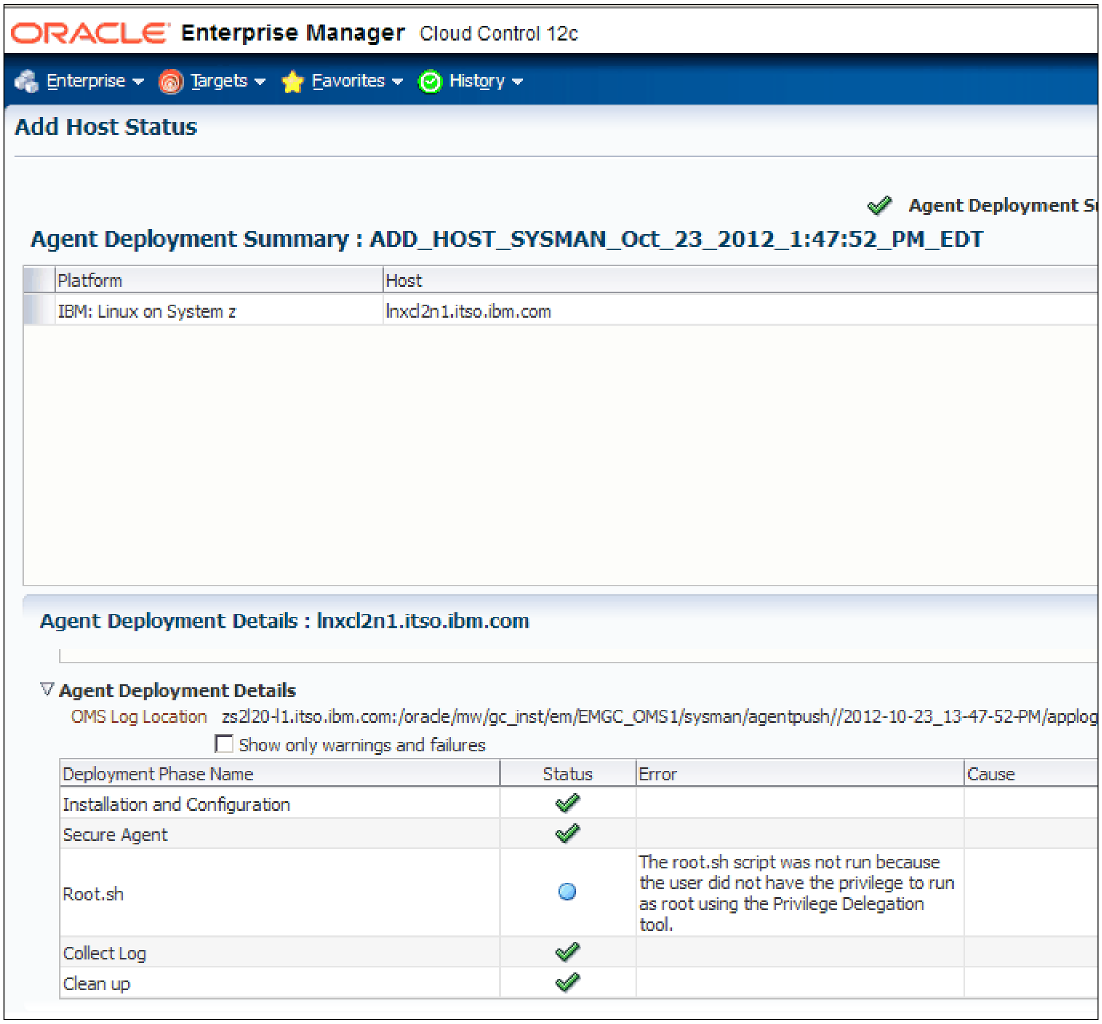


Figure 5-46 Host status

11.Run `root.sh` in the host location (as recommended) and, then click **Done**.

By selecting **Targets** → **Hosts** in the Cloud Control (as shown in Figure 5-47), we can see the availability of the hosts in the Hosts window, as shown in Figure 5-48.

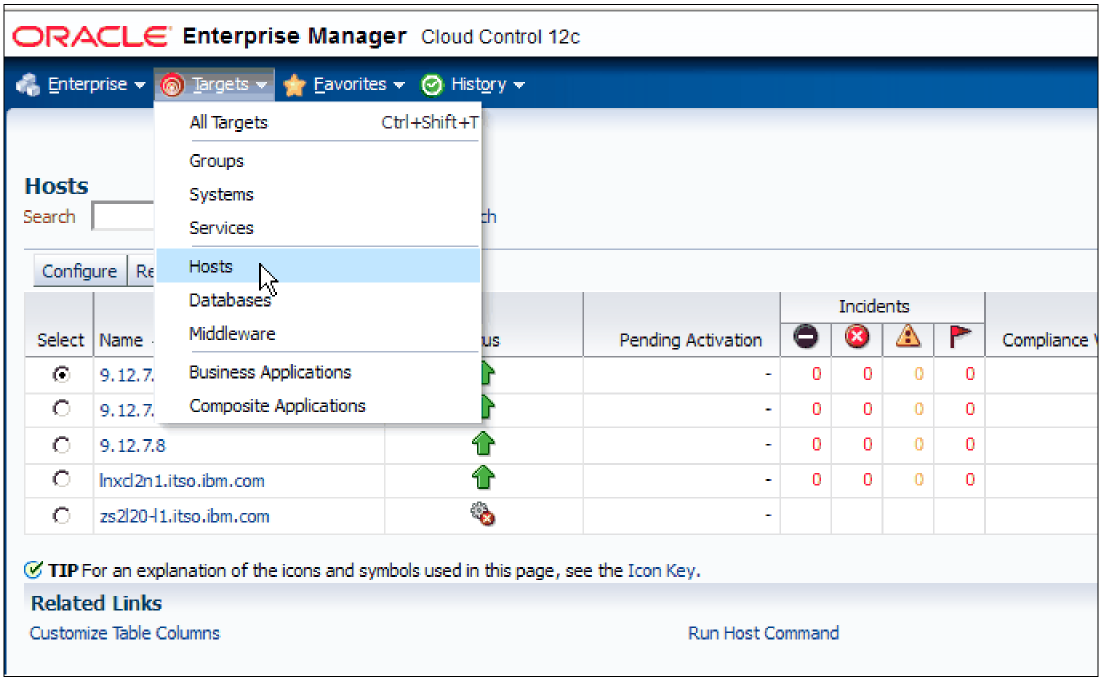


Figure 5-47 Host options

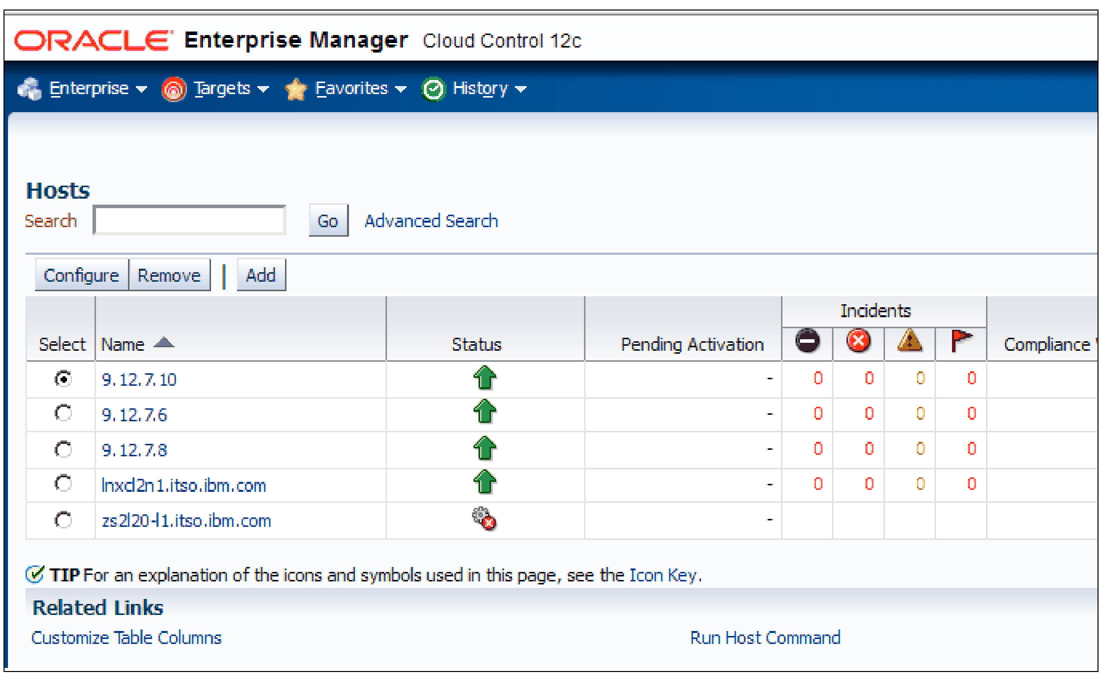


Figure 5-48 Availability of hosts

5.6 Deploying Management Agents in silent mode

Oracle Management Agent 12c for Linux on System z hosts can be deployed from Enterprise Manager Cloud Control 12c console or by using the silent installation method. In this section, we describe the processes to deploy it in silent mode. Installing a Management Agent in silent mode is an alternative to installing it by using the Add Host Target wizard. The silent mode requires the use of a response file for providing the installation details and a deployment script (`agentDeploy.sh`) for silently installing the Management Agent by using the information that is supplied in the response file.

The process that is used to install Oracle Management Agent 12c for Linux on System z in silent mode includes the following steps:

1. At the EM Cloud Control server, download the Agent Software as a `.zip` file.
2. Customize a response file at the Destination host.
3. Use an AgentDeploy script at the Destination host.

In our example, we used the **`emcli`** commands to check the list of supported platforms. After the list of available updates are populated in the system, the required Management Agent software can be downloaded and then applied that to the Destination host by completing the following steps:

1. Check the available Platform Management Agents software by using the following **`emcli`** commands, as shown in Figure 5-49:
   - **`$ emcli login`**
   - **`$ emcli sync`**
   - **`$ emcli get_supported_platforms`**

```
[oracle@zs2120-l1 patches]$ ./emcli login -username=sysman -password=webl0gic
Login successful
[oracle@zs2120-l1 patches]$ ./emcli sync
Synchronized successfully
[oracle@zs2120-l1 patches]$ ./emcli get_supported_platforms
Getting list of platforms ...
Check the logs at /zCode/patches/.emcli/agent.log
About to access self-update code path to retrieve the platforms list..
Getting Platforms list  ...
-----------------------------------------------
Version = 12.1.0.2.0
 Platform = Linux x86-64
-----------------------------------------------
Version = 12.1.0.2.0
 Platform = IBM: Linux on System z
-----------------------------------------------
Platforms list displayed successfully.
[oracle@zs2120-l1 patches]$
```

Figure 5-49 Checking supported platforms

   From the output, you can see that IBM: Linux on System z, 12.1.0.2.0 is available in the Enterprise Manager Cloud Control Server.

2. Run the **$ ./emcli get_agentimage** command to download the Agent software, as shown in Figure 5-50.

```
[oracle@zs2120-l1 patches]$ ./emcli get_agentimage -destination=/zCode/agentdeploy -p
inux on System z" -version=12.1.0.2.0
Platform:IBM: Linux on System z
Destination:/zCode/agentdeploy
 === Partition Detail ===
Space free : 20 GB
Space required : 1 GB
Check the logs at /zCode/patches/.emcli/get_agentimage_2012-10-17_16-46-54-PM.log
Setting property ORACLE_HOME to:/oracle/mw/oms
calling pulloneoffs with arguments:/oracle/mw/oms/zCode/EMGridSWLib/34/CBF65B6959291
8FE12.1.0.2.0linux64_zseries
Check this logs for more information: /oracle/mw/oms/sysman/prov/agentpush/logs
[oracle@zs2120-l1 patches]$ ls -alF /zCode/agentdeploy/
total 290356
drwxr-xr-x. 2 oracle oinstall      4096 Oct 17 16:47 ./
drwxrwxrwx. 8 oracle oinstall      4096 Oct 17 16:45 ../
-rw-r-----. 1 oracle oinstall 297018886 Oct 17 16:47 12.1.0.2.0_AgentCore_209.zip
[oracle@zs2120-l1 patches]$
```

Figure 5-50 Download the file

3. The software needs at least 1 GB of storage. Copy this file into the Destination host and extract the `.zip` there, as shown in Figure 5-51.

```
12.1.0.2.0_AgentCore_209.zip         agentcoreimage.zip  agentimage.properties  unzip
12.1.0.2.0_PluginsOneoffs_209.zip    agentDeploy.sh      agent.rsp
[oracle@zs2120-l1 agentdeploy]$
```

Figure 5-51 Extracted file

At the Destination host, the `agent.rsp` file was customized, as shown in Figure 5-52.

```
#-------------------------------------------------------------------------------
OMS_HOST=9.12.5.131
EM_UPLOAD_PORT=7802
AGENT_REGISTRATION_PASSWORD=webl0gic
AGENT_INSTANCE_HOME=/u01/oracle/agent2
AGENT_PORT=1840
b_startAgent=true
ORACLE_HOSTNAME=9.12.7.10
s_agentHomeName=agent1
#-------------------------------------------------------------------------------
```

Figure 5-52 Customized agent

4. Run the following `agentdeploy.sh` command at the Destination host, as shown in Figure 5-53:

`$ agentDeploy.sh AGENT_BASE_DIR=/u01/oracle/agent2 RESPONSE_FILE=/tmp/foragent/agent912710.rsp`

```
rade=true
Clone Action Logs Location:/u01/grid/oraInventory/logs/cloneActions<timestamp>.log
Cloning of agent home completed successfully

Attaching sbin home...
Executing command: /u01/oracle/agent/core/12.1.0.2.0/oui/bin/runInstaller -ignoreSysP
ome -waitForCompletion -nowait ORACLE_HOME=/u01/oracle/agent/sbin ORACLE_HOME_NAME=sb
Attach Home Logs Location:/u01/oracle/agent/core/12.1.0.2.0/cfgtoollogs/agentDeploy/A
tamp>.log
Attach home for sbin home completed successfully.

Updating home dependencies...
Executing command: /u01/oracle/agent/core/12.1.0.2.0/oui/bin/runInstaller -ignoreSysP
omeDeps -waitForCompletion HOME_DEPENDENCY_LIST={/u01/oracle/agent/sbin:/u01/oracle/a
0.2.0,} -invPtrLoc /u01/oracle/agent/core/12.1.0.2.0/oraInst.loc -force
Update Home Dependencies Location:/u01/oracle/agent/core/12.1.0.2.0/cfgtoollogs/agent
meDeps<timestamp>.log
Update home dependency completed successfully.

Performing the agent configuration...
Executing command: /u01/oracle/agent/core/12.1.0.2.0/oui/bin/runConfig.sh ORACLE_HOME
ent/core/12.1.0.2.0 RESPONSE_FILE=/u01/oracle/agent/core/12.1.0.2.0/agent.rsp ACTION=
perform COMPONENT_XML={oracle.sysman.top.agent.11_1_0_1_0.xml} RERUN=true
Configuration Log Location:/u01/oracle/agent/core/12.1.0.2.0/cfgtoollogs/cfgfw/CfmLog
log
Agent Configuration completed successfully

The following configuration scripts need to be executed as the "root" user.
#!/bin/sh
#Root script to run
 /u01/oracle/agent/core/12.1.0.2.0/root.sh
To execute the configuration scripts:
1. Open a terminal window
2. Log in as "root"
3. Run the scripts
Agent Deployment Successful.
Agent deployment log location:
/u01/oracle/agent/core/12.1.0.2.0/cfgtoollogs/agentDeploy/agentDeploy_2012-10-17_18-2
Agent deployment completed successfully.
oracle@lnxc2n1:/u01/oracle/tmp>
```

Figure 5-53 Running the Deploy script

This completes the Agent Deployment in silent mode process. From the Grid Control Server Console, we can add the databases that are running on the Destination host to be monitored.

5.7 Adding the databases for monitoring

After the agent for Linux on System z is deployed, we can discover the Oracle Databases on that host for monitoring. Complete the following steps:

1. In Cloud Control, select **Targets** → **Databases**, and then click **Add** in the Databases section, as shown in Figure 5-54.

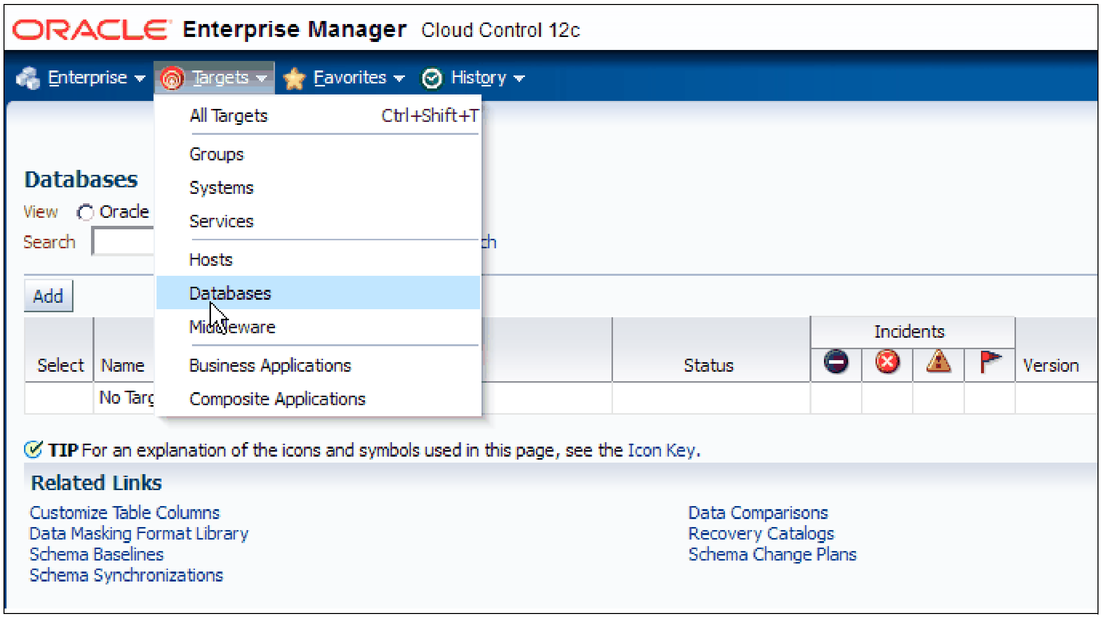


Figure 5-54 Discover the database target

2. In the Host field of the Add Database Instance Target: Specify Host window, specify the fully qualified host name `lnxcl2n1.itso.ibm.com` and click **Continue**, as shown in Figure 5-55.

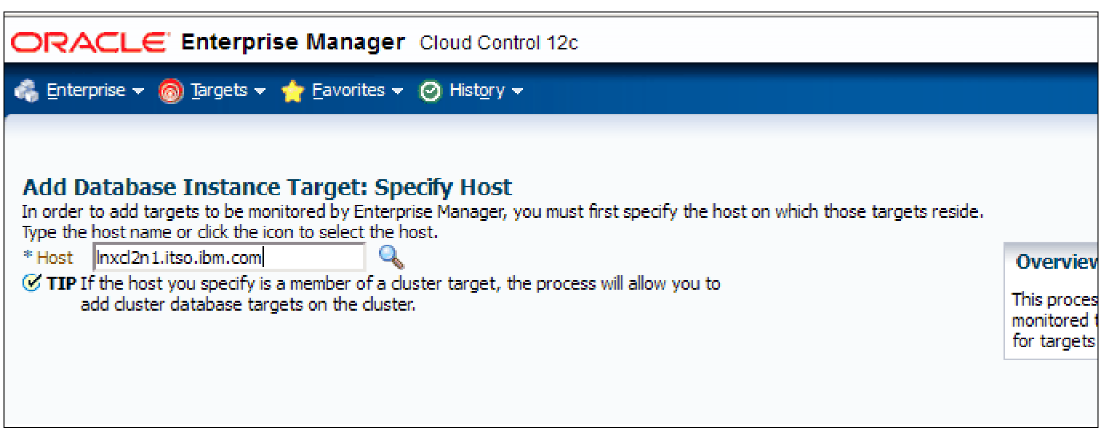


Figure 5-55 Specify the Host name

3. The agent discovers the database (`REMOTEDB`), ASM instance, and the Listener as shown in Figure 5-56.

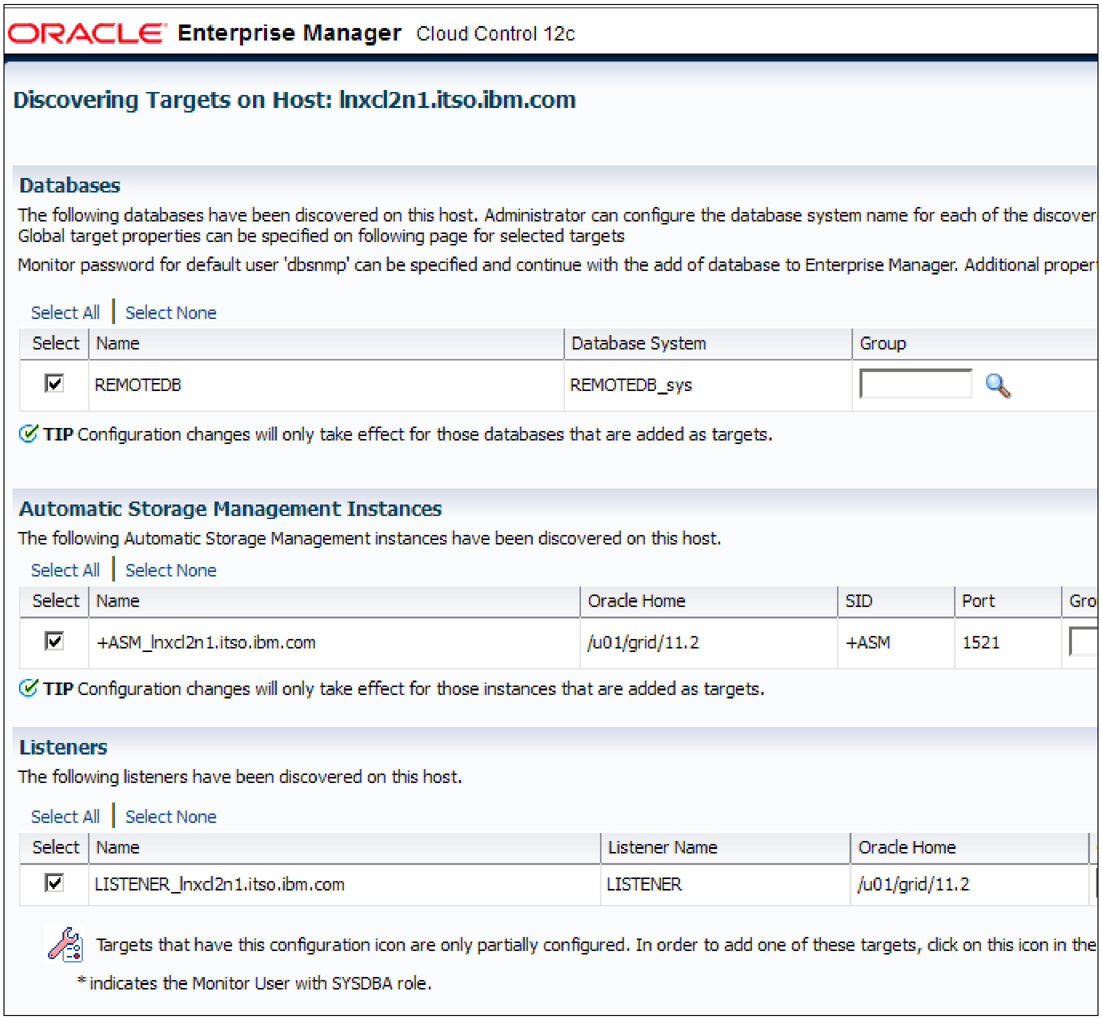


Figure 5-56 Showing the listener

4. Configure the database by selecting the **Configure** at the `REMOTEDB` line and specify the database- related parameters. Save the configuration.

The database configuration is now saved and the databases that are running on Linux on System z in the Oracle Enterprise Manager Cloud Control can be monitored.

5.8 Summary

In this chapter, we shared our experiences with installing a Cloud Control Server on x86 based Linux server and deployed the agents from there to monitor the databases that are running on Linux on System z.

Before deploying the agents, we also updated the Cloud Control Software Library with the required levels of agent software and plug-ins by connecting to the Oracle repository site online.

Starting with Oracle Enterprise Manager Cloud Control 12c Release 2 (12.1.0.2), we showed how the command line utility **`emcli`** can also be used to update the Grid Control Software Library in offline mode. The agents were deployed from the Cloud Control Console and the silent agent deployment option was used at the Linux guests.

We also showed how to enable the cloud control to monitor Oracle databases.

Part 2

Managing an Oracle environment on Linux on System z

In this part, we provide information about the following topics to help you manage your environment:

- Using z/VM new feature of Live Guest Relocation (LGR) to move an Oracle Linux guest from one LPAR to another.

  Important: This feature is not certified by Oracle at the time of this writing. For more information, see Chapter 6, “Using z/VM Live Guest Relocation to relocate a Linux guest” on page 117.

- Considerations for tuning your environment to improve performance of your Oracle database. It covers z/VM, Linux, and Oracle tuning possibilities, as described in Chapter 7, “Tuning z/VM, Linux, and Oracle to run on IBM System z” on page 137.
- Options that were used to migrate Oracle Database instances to an Linux on System z platform, as described in Chapter 8, “Cross-platform migration overview” on page 155.
- Options that can be used to provide High Availability and Disaster Recovery solutions when Oracle Database is run on Linux on System z. These options include Oracle components and IBM components to provide a highly available environment, as described in Chapter 9, “High Availability and Disaster Recovery environment for Oracle” on page 179.

6

Using z/VM Live Guest Relocation to relocate a Linux guest

Attention: The following command conventions are used in this chapter:

- z/VM commands are prefixed with ==>
- Linux commands that are running as root are prefixed with #
- Linux commands that are running as non-root are prefixed with $

The last release of IBM z/VM 6.2 introduced a major feature named Live Guest Relocation (LGR). This chapter provides information about using LGR in an Oracle environment through two examples by relocating the following components:

- An Oracle Single Instance
- An Oracle RAC cluster

This chapter includes the following topics:

- Overview of z/VM 6.2 features
- Lab environment
- Scenario 1: Relocating an active Oracle single-instance database by using LGR
- Scenario 2: Relocating an Oracle RAC inactive node by using LGR

6.1 Overview of z/VM 6.2 features

IBM z/VM 6.2 introduces significant changes to z/VM in the form of multi-system clustering technology, which allows up to four z/VM instances in a single system image (SSI) cluster. This technology is important because it offers you an attractive alternative to vertical growth by adding new z/VM systems. In the past, this capability required duplicate efforts to install, maintain, and manage each system. With SSI, these duplicate efforts are reduced or eliminated.

Support for LGR allows you to move Linux virtual servers without disruption to the business, which helps you to avoid planned outages. The z/VM systems are aware of each other and can use their combined resources. LGR enables you to avoid loss of service because of planned outages by relocating guests from a system that requires maintenance to a system that remains active during the maintenance period. The SSI and LGR technologies offer substantial value and are a major departure from past z/VM practices.

6.1.1 Single System Image feature

The z/VM Single System Image feature (VMSSI) is an optionally priced feature that is new with z/VM Version 6.2. It enables up to four z/VM systems to be configured as members of an SSI cluster, which share the following resources:

- User directory
- DASD volumes
- User minidisks
- Spool files
- Network devices

Members can be on the same or separate Central Processor Complexes (CPCs). They can be first-level or second-level z/VM systems. SSI enables the members of the cluster to be managed as one system, which allows service to be applied to each member of the cluster, thus avoiding an outage to the entire cluster.

6.1.2 LGR

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

You might need to relocate a running virtual server for the following reasons:

- Maintaining hardware or software
- Fixing performance problems
- Rebalancing workload

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

LGR support: Linux on System z is the only guest environment that is supported by LGR. Because the LGR process is not yet certified by Oracle, it is not recommended for use in relocating an active Oracle RAC node.

In this chapter, we describe two scenarios of Oracle relocation between two members of a z/VM SSI cluster with an Oracle Single Instance and an Oracle two nodes RAC by stopping each node before their relocations.

6.2 Lab environment

In our lab environment, we set up one z/VM SSI cluster with two members to demonstrate the process of relocation of Oracle environment between these members by using LGR functionality.

6.2.1 Overview of tested two node z/VM SSI cluster

The SSI cluster is named `MOPVMSSI`. It is composed of two members: `MOPVMEM1` and `MOPVMEM2`, as shown on Figure 6-1.

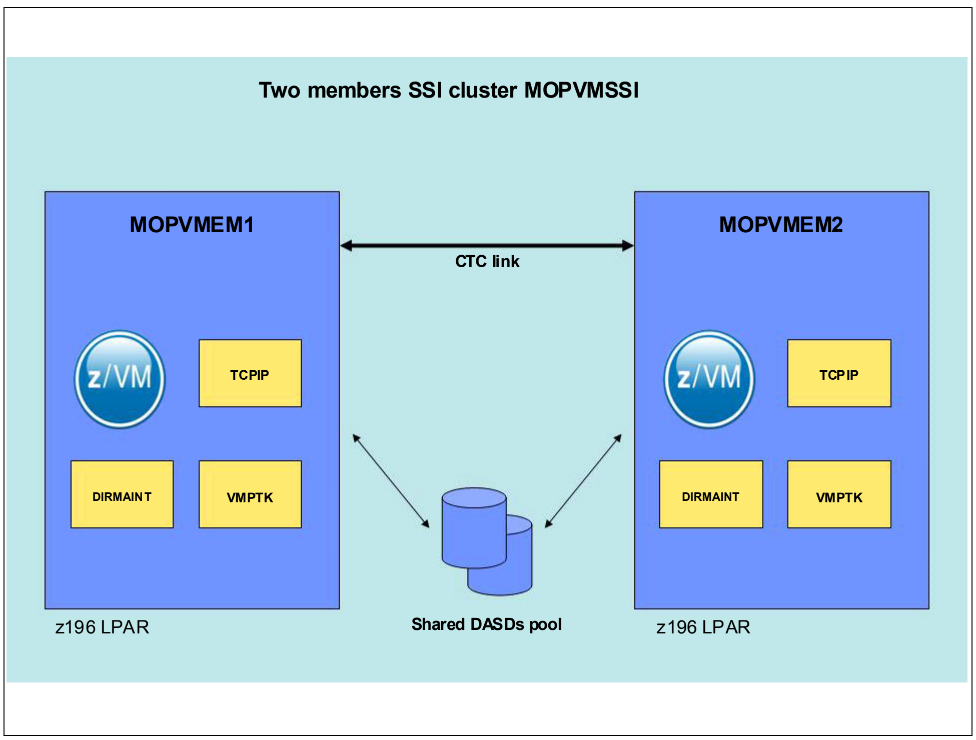


Figure 6-1 Overview of tested z/VM SSI cluster

Use CP command **QUERY SSI** to display the information about your SSI cluster, as shown in the following example:

```
==> query SSI
SSI Name: MOPVMSSI
SSI Mode: Stable
Cross-System Timeouts: Enabled
SSI Persistent Data Record (PDR) device: VMPCOM on 70BE
SLOT SYSTEMID STATE     PDR HEARTBEAT       RECEIVED HEARTBEAT
   1 MOPVMEM1 Joined    11/28/12   18:08:07 11/28/12   18:08:07
   2 MOPVMEM2 Joined    11/28/12   18:08:16 11/28/12   18:08:16
   3 -------- Available
   4 -------- Available
Ready; T=0.01/0.01 18:08:28
```

6.2.2 Hardware configuration

We used an IBM z196 to host the two LPARs. Each logical partition (LPAR) has 32 GB of memory. We configured the channel-to-channels (CTCs) between each LPAR using inter-system facility for communications (ISFC) as a prerequisite for SSI cluster connection. Each LPAR is connected to the network by using an open system adapter (OSA). Equivalency identifiers (EQIDs) are assigned to the OSAs as a prerequisite for LGR. We shared a pool of direct access storage devices (DASDs) between the two LPARs to host our Oracle environment.

For more information about how to define z/VM resources in your SSI cluster to perform LGR, see *An Introduction to z/VM Single System Image (SSI) and Live Guest Relocation (LGR),* SG24-8006.

6.2.3 z/VM Software

We installed and customized the following products:

- TCP/IP stack
- IBM Directory Maintenance (DirMaint™)
- Performance Toolkit for z/VM (PERFTK)

6.2.4 Solution that is used to simulate a database workload

During our scenarios, we simulate the client workload on the Oracle database with Swingbench tool (see running in a Linux on System z guest.

For more information about the Swingbench tool, see this website:

http://www.dominicgiles.com/swingbench.html

6.3 Scenario 1: Relocating an active Oracle single-instance database by using LGR

In this scenario, we describe the steps that are used to relocate an Oracle Single-instance database that is running in a Linux on System z guest between members of a z/VM SSI cluster.

To build this scenario, we installed and set up a Single Instance Oracle 11gR2 database. Figure 6-2 shows the tested infrastructure.

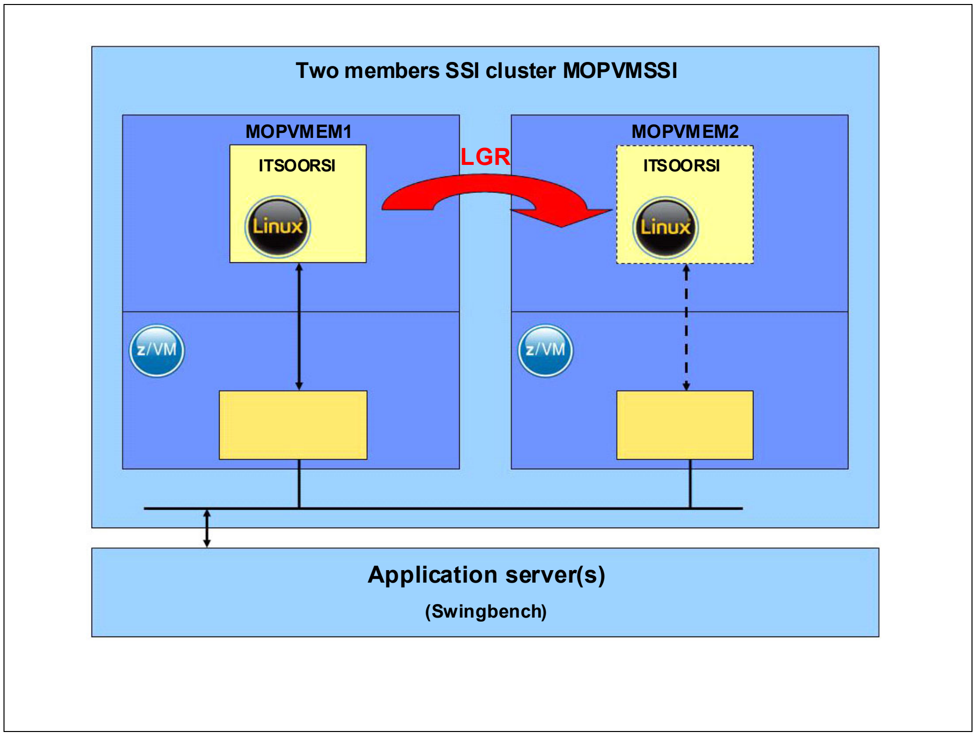


Figure 6-2 Infrastructure for Scenario 1

6.3.1 Setup information

This environment is installed in a single SUSE Linux Enterprise Server 11 SP1 named `ITSOORSI` with 8 GB of memory and two virtual processors.

Table 6-1 describes the disk layout of the `ITSOORSI` guest.

Table 6-1 ITSOORSI Disk layout

Virtual Device address	Disk type	Size (MB)	Mount
100	3390 mod 9	7042	/
101	VDisk	146	First level swap
102	3390 mod 9	7042	Second-level swap
200	3390 mod 9	7042	*/u01/oracle*

ITSOORSI is connected to an OSA adapter through a layer 2 virtual switch (VSWITCH) SWCLO. The SWCLO VSWITCH was defined identically in both members of the SSI cluster to allow the LGR.

As a requirement for a guest to by eligible for LGR, we add the option CHPIDVIRTUALIZATION ONE in the user directory, as shown in the following example:

```
USER ITSOORSI XXXXXX 8G 8G G
    COMMAND COUPLE 0D20 SYSTEM SWCLO
    COMMAND DEFINE VFB-512 AS 0101 BLK 299008 1
    CPU 00 BASE
    CPU 01
    MACHINE ESA 4
    OPTION CHPIDVIRTUALIZATION ONE 2
    SCR INA WHI NON STATA RED NON CPOUT YEL NON VMOUT GRE NON INRED TUR
    CONSOLE 0009 3215 T
    NICDEF 0D20 TYPE QDIO DEVICES 3 LAN SYSTEM SWCLO 3
    SPOOL 000C 2540 READER *
    SPOOL 000D 2540 PUNCH A
    SPOOL 000E 1403 A
    MDISK 0100 3390 1 10016 CL2B03 MR 4
    MDISK 0102 3390 1 10016 CL2B04 MR 5
    MDISK 0200 3390 1 10016 CL2B05 MR 6
```

The definition of ITSOORSI virtual machine includes the following entries (the numbers in the following list refer to the numbers in the preceding example):

1. First-level swap on Virtual Disk.
2. Sets CHPID virtualization to ONE.
3. Virtual network adapter that is connected to virtual switch SWCLO definition.
4. Minidisk that is used for root.
5. Minidisk that is used for Oracle binaries.
6. Second-level swap on minidisk.

6.3.2 Simulating the client workload

The client workload is simulated with Swingbench running in a Linux on System z guest that is connected on the same network as ITSOORSI.

To configure the access to the database with Oracle Call Interface (OCI), we use the tsnnames.ora file, as shown in the following example:

```
$ cat tnsnames.ora
# tnsnames.ora Network Configuration File:
/u01/oracle/product/11.2.0/dbhome_1/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

ITSO =                                        1
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.3.58.50)(PORT = 1521))    2
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ITSO)                    3
    )
  )
```

This file is used by Oracle clients to retrieve the following information that is required to connect to the database (the numbers in the following list refer to the numbers in the preceding example):

1. Connection name
2. Listener address of the database
3. Database service name

In Swingbench GUI that is shown in Figure 6-3, we define the connection information to access the database (the service name as defined in the `tnsnames.ora` file), the number of users to generate, and other benchmark runtime parameters.

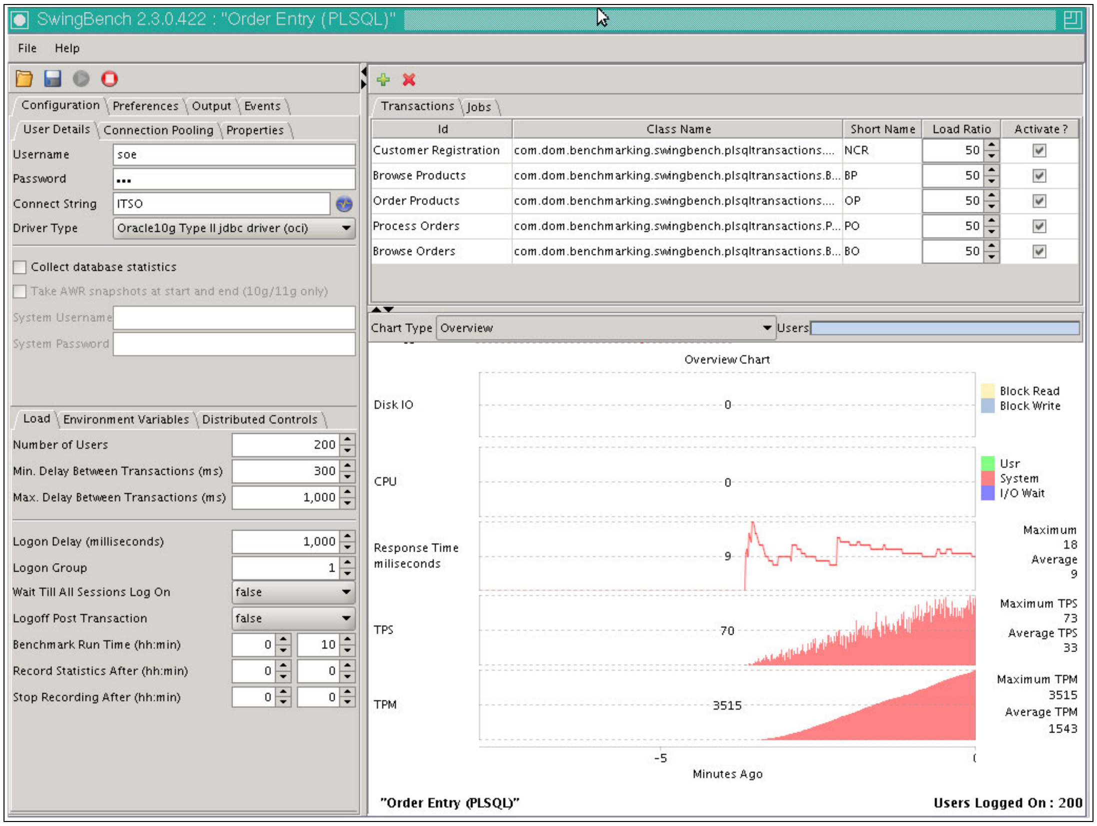


Figure 6-3 Swingbench GUI with parameters used for Scenario 1

We completed the following steps to verify the connectivity between Swingbench tool and Oracle database check:

1. TCP sockets were opened between Swingbench and the Oracle database guest `ITSOROSI`, as shown in the following example:

```
# netstat -a | grep swingbench | grep 10.3.58.50
tcp        0      0 swingbench.:60814 10.3.58.50:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:60667 10.3.58.50:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:60631 10.3.58.50:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:60769 10.3.58.50:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:60731 10.3.58.50:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:60776 10.3.58.50:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:60749 10.3.58.50:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:60652 10.3.58.50:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:60700 10.3.58.50:ncube-lm     ESTABLISHED
```

2. TCP sockets were opened between Swingbench and ITSOROSI, as shown in the following example:

```
# netstat -a | grep swingbench | grep 10.3.58.50 | wc -l
200
```

A total of 200 connections are opened, which correspond to the 200 users that are defined in Swingbench.

6.3.3 Relocating an Oracle guest

Class A user needed: We run the z/VM commands from a class A user ID (MAINT for example) of MOPVMEM1

Complete the following steps to perform guest relocation:

1. Check ITSOORSI is running in MOPVMEM1, as shown in the following example:

```
==> QUERY ITSOORSI AT ALL
MOPVMEM1 : ITSOORSI - DSC
```

2. Test if it is eligible for relocation to MOPVMEM2, as shown in the following example:

```
==> VMRELOCATE TEST ITSOORSI TO MOPVMEM2
User ITSOORSI is eligible for relocation to MOPVMEM2
Ready; T=0.01/0.01 17:51:24
```

3. Perform the effective relocation to MOPVMEM2, as shown in the following example:

```
==> VMRELOCATE MOVE ITSOORSI TO MOPVMEM2
Relocation of ITSOORSI from MOPVMEM1 to MOPVMEM2 started
User ITSOORSI has been relocated from MOPVMEM1 to MOPVMEM2
Ready; T=0.01/0.01 17:52:31
```

4. Verify that the relocation was successful, as shown in the following example:

```
==> QUERY ITSOORSI AT ALL

MOPVMEM2 : ITSOORSI - DSC
```

We can see the quiesce time of the guest by running a ping command to the gateway each second during the relocation process. In our test, ITSOORSI is freezing for about 6 seconds (between 16:55:03 and 16:55:10), as shown in the following example:

```
# ping 10.3.58.254 | awk '/64/ {"date" | getline date ; print $0, "\t\t" date ; close("date")}'
64 bytes from 10.3.58.254: icmp_seq=54 ttl=255 time=0.538 ms           Mon Dec 3 16:55:01 CET 2012
64 bytes from 10.3.58.254: icmp_seq=55 ttl=255 time=7.77 ms            Mon Dec 3 16:55:02 CET 2012
64 bytes from 10.3.58.254: icmp_seq=56 ttl=255 time=0.499 ms           Mon Dec 3 16:55:03 CET 2012
64 bytes from 10.3.58.254: icmp_seq=58 ttl=255 time=0.561 ms           Mon Dec 3 16:55:10 CET 2012
64 bytes from 10.3.58.254: icmp_seq=59 ttl=255 time=0.590 ms           Mon Dec 3 16:55:11 CET 2012
64 bytes from 10.3.58.254: icmp_seq=60 ttl=255 time=0.797 ms           Mon Dec 3 16:55:12 CET 2012
```

In the Swingbench GUI, we can observe that during the quiesce time, the transactions froze (as shown in Figure 6-4) in the Transaction per seconds graph. However, all of the users remained connected to the database.

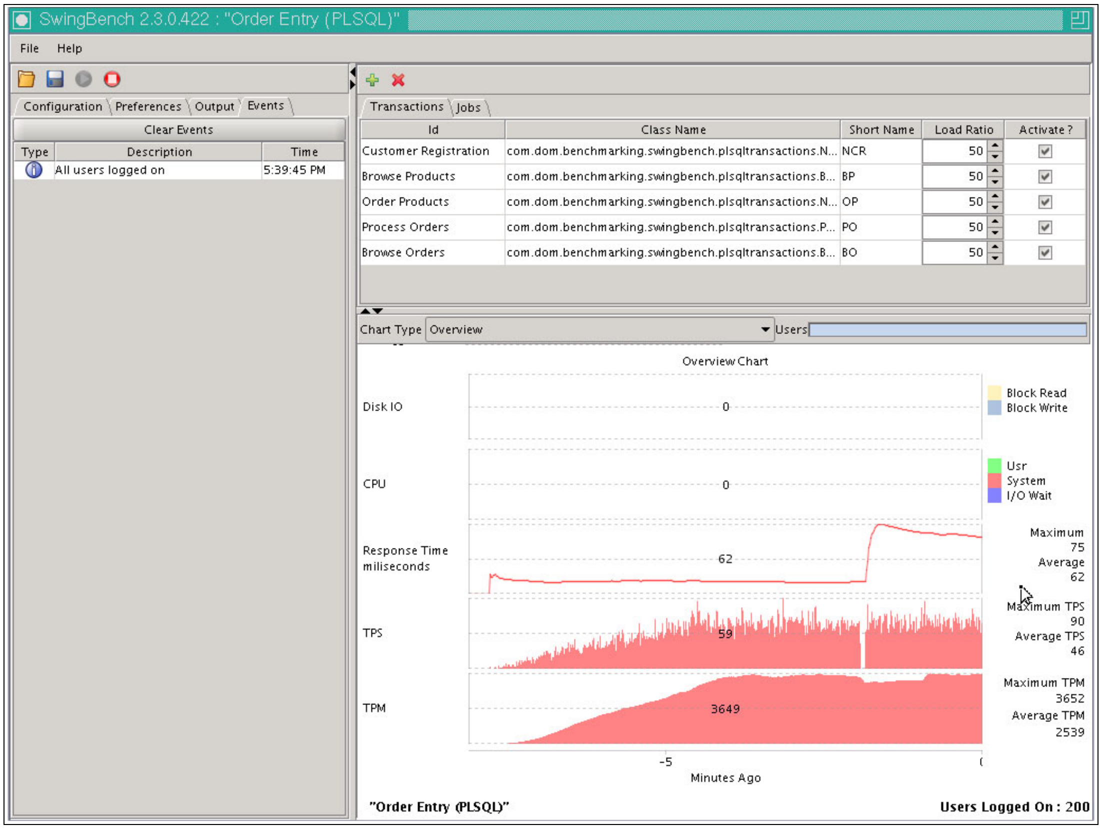


Figure 6-4 Swingbench GUI during relocation in Scenario 1

6.4 Scenario 2: Relocating an Oracle RAC inactive node by using LGR

In this section, we describe the steps that are used to relocate a two nodes Oracle RAC environment running in Linux guests between members of a z/VM SSI cluster.

Because the Live Guest Relocation on an active RAC node was not certified yet, complete the following steps:

1. Stop the Oracle cluster process on one of the nodes (leaving the Linux active). The node leaves the Oracle cluster.
2. Relocate this Linux guest to the other z/VM using LGR.
3. Restart the Oracle cluster on the Linux guest. It rejoins the Oracle cluster.
4. Stop the Oracle cluster process on the second node.
5. Relocate it to the other z/VM.
6. Restart Oracle cluster processes on the second Linux guest.

To build this scenario, we installed and set up a two-node 11g R2 Oracle RAC database and configured it with Oracle application failover technology to provide continuous operation of the database and balancing the client connection to the other node during relocation.

When one of the nodes is shut down, the clients are automatically reconnected to the survival node. Figure 6-5 shows the tested infrastructure.

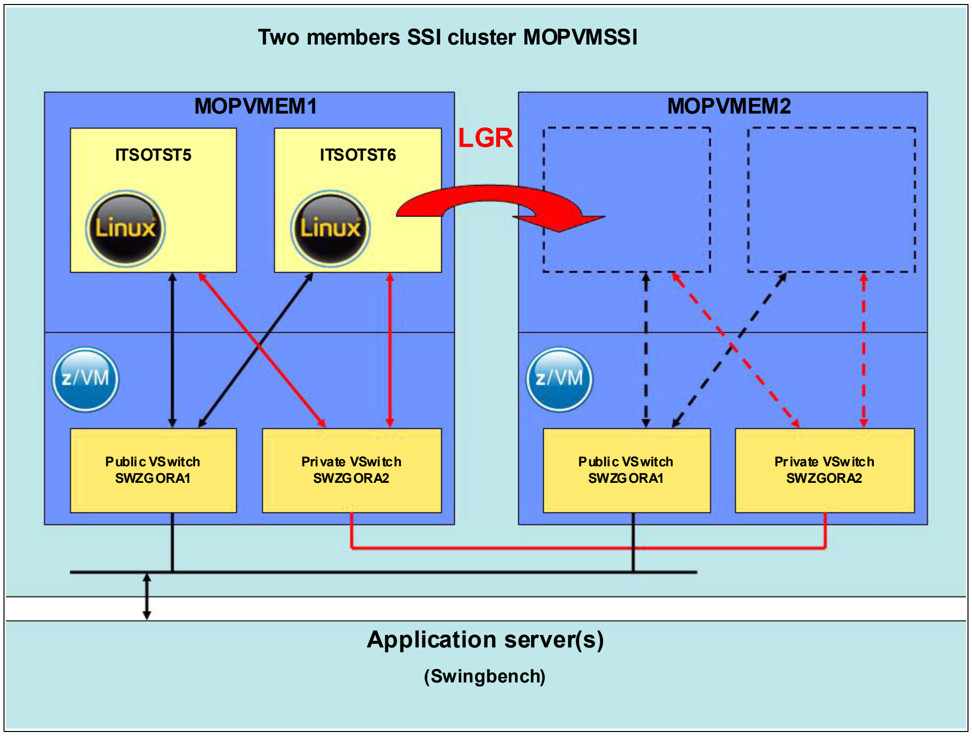


Figure 6-5 Infrastructure for Scenario 2

6.4.1 Setup information

We defined two Linux virtual machines, `ITSOTST5` and `ITSOTST6`, to host the two nodes of Oracle RAC. Both machines were configured with 8 GB of memory, two virtual processors, and ran SUSE Linux Enterprise Server 11 SP1, as shown in the following example:

```
USER ITSOTST5 XXXXXX 8G 8G G
   COMMAND COUPLE 0D40 SYSTEM SWZGORA1
   COMMAND COUPLE 0D50 SYSTEM SWZGORA2
   COMMAND DEFINE VFB-512 AS 0101 BLK 299008                    1
   CPU 00 BASE
   CPU 01
   MACHINE ESA 4
   OPTION CHPIDV ONE                            2
   SCR INA WHI NON STATA RED NON CPOUT YEL NON VMOUT GRE NON INRED TUR
   CONSOLE 0009 3215 T
   NICDEF 0D40 TYPE QDIO DEVICES 3 LAN SYSTEM SWZGORA1      3
   NICDEF 0D50 TYPE QDIO DEVICES 3 LAN SYSTEM SWZGORA2      3
   SPOOL 000C 2540 READER *
   SPOOL 000D 2540 PUNCH A
   SPOOL 000E 1403 A
   MDISK 0100 3390 1 10016 CL2B06 MR      4
   MDISK 0102 3390 1 10016 CL2B07 MR      5
   MDISK 0200 3390 1 10016 CL2B08 MR      6
   MDISK 0400 3390 1 10016 CL2B09 MW     7
   MDISK 0401 3390 1 10016 CL2B0A MW    7
   MDISK 0402 3390 1 10016 CL2B0B MW    7
```

```
  MDISK 0300 3390 1 10016 CL2B0C W       8
  MDISK 0301 3390 1 10016 CL2B0D W       8
  MDISK 0302 3390 1 10016 CL2B0E W       8
  MDISK 0303 3390 1 10016 CL2B0F W       8
  MDISK 0304 3390 1 10016 CL2B10 W       8

USER ITSOTST6 XXXXXX 8G 8G G
  COMMAND COUPLE 0D40 SYSTEM SWZGORA1
  COMMAND COUPLE 0D50 SYSTEM SWZGORA2
  COMMAND DEFINE VFB-512 AS 0101 BLK 299008                        1
  CPU 00 BASE
  CPU 01
  MACHINE ESA 4
  OPTION CHPIDV ONE             2
  SCR INA WHI NON STATA RED NON CPOUT YEL NON VMOUT GRE NON INRED TUR
  CONSOLE 0009 3215 T
  NICDEF 0D40 TYPE QDIO DEVICES 3 LAN SYSTEM SWZGORA1                3
  NICDEF 0D50 TYPE QDIO DEVICES 3 LAN SYSTEM SWZGORA2                3
  SPOOL 000C 2540 READER *
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
  LINK ITSOTST5 0400 0400 MW              7
  LINK ITSOTST5 0401 0401 MW              7
  LINK ITSOTST5 0402 0402 MW              7
  MDISK 0100 3390 1 10016 CL2B11 MR     4
  MDISK 0102 3390 1 10016 CL2B12 MR     5
  MDISK 0200 3390 1 10016 CL2B13 MR     6
  MDISK 0300 3390 1 10016 CL2B14 W       8
  MDISK 0301 3390 1 10016 CL2B15 W       8
  MDISK 0302 3390 1 10016 CL2B16 W       8
  MDISK 0303 3390 1 10016 CL2B17 W       8
  MDISK 0304 3390 1 10016 CL2B10 W       8
```

The definition of ITSOTST5 and ITSOTST6 virtual machines includes the following entries (the following numbers correspond to the numbers in the preceding example):

1. First-level swap on Virtual Disk.
2. Sets CHPID virtualization to ONE.
3. Virtual network adapters definition.
4. Minidisk that is used for root file system.
5. Second-level swap on minidisk.
6. Minidisk that is used for Oracle binaries.
7. Disks that are used for Oracle Grid installation (for ITSOTST6, these are links to ITSOTST5 disks).
8. Disks that are used by ASM.

By using this infrastructure, we installed Oracle RAC with an Oracle GRID infrastructure (Oracle ASM and Oracle Clusterware) to provide a High Availability (HA) environment.

For more information about High Availability, see Chapter 9, “High Availability and Disaster Recovery environment for Oracle” on page 179.

6.4.2 Network configurations

Various network configurations are certified for an Oracle RAC running under z/VM. For more information, see Chapter 3, “Network connectivity options for Oracle on Linux on IBM System z” on page 29.

In Scenario 2, we choose to create the following virtual switches (vSwitch) in each z/VM:

- Public vSwitch `SWZGORA1` to access the RAC from outside
- Private vSwitch `SWZGORA2` for the interconnect network

> **Tip:** Layer 2 HiperSockets are supported for LGR. If your SSI members run in the same physical machine, we recommend the use of a HiperSockets interface instead of a VSWITCH for the private interconnect network to use the unique network architecture of IBM System z.
>
> For more information, see Chapter 9, “High Availability and Disaster Recovery environment for Oracle” on page 179.

An Oracle RAC environment requires a specific Linux network configuration. Each node needs to have at least the following IP addresses:

- Public interface IP used to reach the Linux guest through SSH and perform installation, configuration, management, and maintenance of the Operating System and Oracle RAC node.
- Private interconnect interface IP that is used for the communication between the nodes of the RAC. Generally, this connection is on a private and non-routable network.
- Oracle Virtual IP (VIP) defined on the same subnetwork as the public IP. This interface is automatically created during the Oracle RAC installation.

With Oracle RAC 11g, you can configure Single Client Access Name (SCAN) listener, which provides a unique name to connect to the cluster database. The client connection descriptor contains only the SCAN listener name and not all the local listener addresses of each node. Oracle automatically redirects the client connection to an available node. If a node fails or is unreachable, Oracle redirects clients’ connections to a healthy node. This functionality is a part of the Oracle load balancing and failover capabilities. Oracle recommends to set up a DNS SCAN with three SCAN IP addresses defined on it and to add an entry for this DNS in the `/etc/host` file of each node. In our demonstration environment, we do not set up a DNS to provide the SCAN IPs. Instead, we provide a single SCAN IP directly on the host file of each node.

We set up the network configuration of ITSOTST5 as shown in Example 6-1.

Example 6-1 Network configuration of ITSOTST5

```
itsotst5:~ # cat /etc/hosts
#itsotst5 public IP
10.3.58.51 itsotst5.mop.ibm.com itsotst5
#itsotst5 interconnect IP
10.7.17.10 itsotst5-priv
#itsotst5 virtual IP
10.3.58.55 itsotst5-vip
#itsotst6 public IP
10.3.58.52 itsotst6.mop.ibm.com itsotst6
#itsotst6 interconnect IP
10.7.17.11 itsotst6-priv
#itsotst5 private IP
10.3.58.56 itsotst6-vip
#SCAN listener IP
10.3.58.57 itsotst
```

We configured the SCAN listener ITSOTST with the IP address 10.3.58.57, and the nodes as described in Table 6-2.

Table 6-2 Network configuration of the RAC

Hostname	Public IP	Virtual IP	Interconnect IP
ITSOTST5	10.3.58.51	10.3.58.55	10.7.17.10
ITSOTST6	10.3.58.52	10.3.58.56	10.7.17.11

6.4.3 Client configuration of Transparent Application Failover

Before relocating a Linux node, we have to shut down the Oracle processes that are running on it. To ensure that all of the users that are connected to the falling node do not lose their connections to the database, we have to provide connection recovery capabilities. For more information about HA solutions on Oracle, see Chapter 9, “High Availability and Disaster Recovery environment for Oracle” on page 179.

In Scenario 2, we used the Transparent Application Failover (TAF). This solution provides client recovery capabilities in case of failed connection by restarting the query in a healthy node. If the failure happens during a transaction, it is rolled back automatically.

The TAF is configured on the client side (in our environment, this is where Swingbench is running) in the tnsnames.ora file. We enter network information and a database service name to reach the database from the client side and different information for the TAF configuration, as shown in the following example:

```
$ cat /u01/oracle/product/11.2.0/dbhome_1/network/admin/tnsnames.ora
ITSOCLUSTER =                                  1
  (DESCRIPTION =
    (LOAD_BALANCE = on) 2
    (FAILOVER = on)
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.3.58.57)(PORT = 1521))    3
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ITSORAC)          4
```

```
            (FAILOVER_MODE=(TYPE=SELECT)(METHOD=BASIC))          5
        )
    )
```

This example included the following components (the following numbers correspond to the numbers that are shown in the preceding example):

1. The connection name.
2. By setting LOAD_BALANCE=on, Oracle RAC distributes the users connection fairly across the different nodes accessing the same database.
3. The address of the SCAN listener.
4. The database service name.
5. The definition of the FAILOVER_MODE to provide client recovery capabilities when a node is shut down.

> **Tip:** Make sure that the remaining node (or nodes) can handle all of the connections that are redirected by the TAF from the failed node.

6.4.4 Simulating the client workload

The client workload is simulated with Swingbench running in a Linux on System z guest connected on the same network as the Oracle RAC nodes. In the Swingbench GUI that is shown in Figure 6-6, we define the connection information to access the database (the service name as defined in the `tnsnames.ora` file), the number of users to generate, and other benchmark runtime parameters.

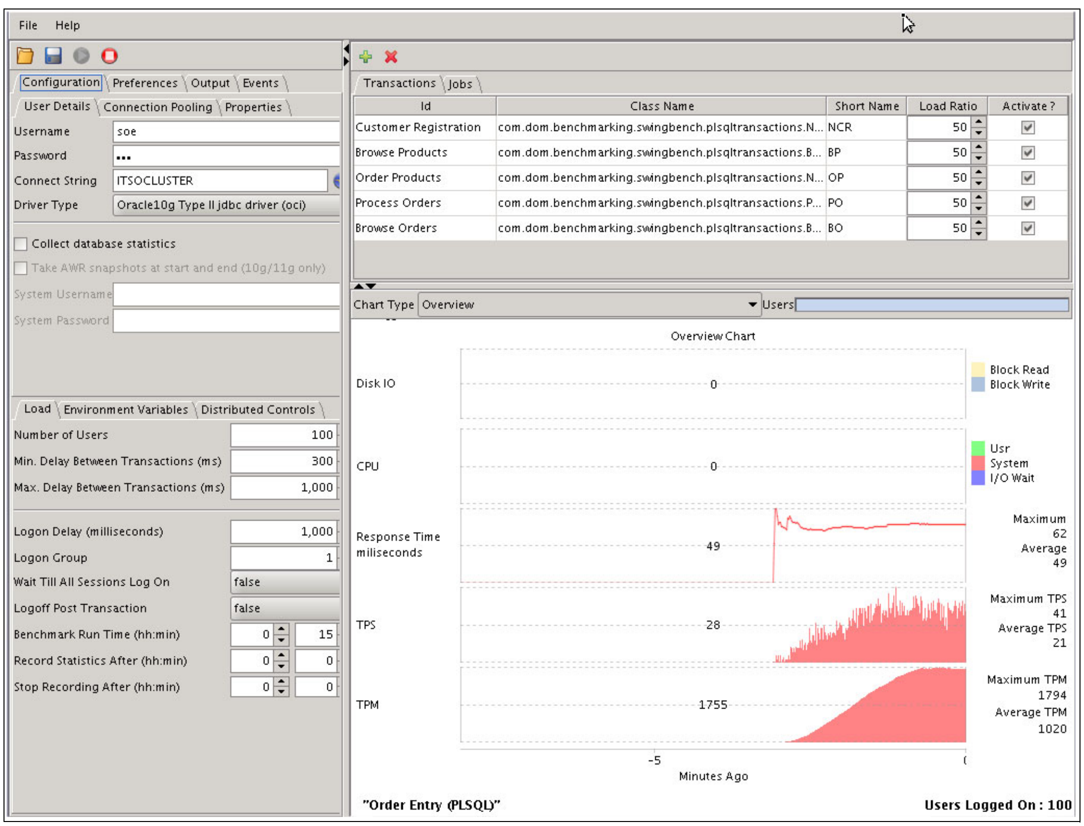


Figure 6-6 Swingbench GUI with parameters used for Scenario 2

In accordance with our TAF configuration, Swingbench generates the workload of 100 users using the single IP address of the SCAN Listener and RAC balances the connection between the two nodes.

To verify the connectivity between Swingbench tool and Oracle RAC, check the following components:

- The state of the nodes, as shown in the following example:

```
itsotst5:~ # su - grid
grid@itsotst5:/home/grid> crsctl status server
NAME=itsotst5
STATE=ONLINE

NAME=itsotst6
STATE=ONLINE
```

- The TCP sockets that are opened between Swingbench and the Oracle RAC, as shown in the following example:

```
swingbench:~ # netstat -a | grep swingbench | grep 10.3.58.5
tcp        0      0 swingbench.:46789 10.3.58.56:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:46777 10.3.58.56:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:59034 10.3.58.55:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:59030 10.3.58.55:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:59042 10.3.58.55:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:59038 10.3.58.55:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:59026 10.3.58.55:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:46785 10.3.58.56:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:46781 10.3.58.56:ncube-lm     ESTABLISHED
tcp        0      0 swingbench.:46773 10.3.58.56:ncube-lm     ESTABLISHED
```

- The number of TCP sockets that are opened between Swingbench and `ITSOTST5`, as shown in the following example:

```
swingbench:~ # netstat -a | grep swingbench | grep 10.3.58.55 | wc -l
50
```

  There are 50 connections that are open, which corresponds to the half of the 100 users that are defined in Swingbench.

- The number of TCP sockets that are opened between Swingbench and `ITSOTST6`, as shown in the following example:

```
swingbench:~ # netstat -a | grep swingbench | grep 10.3.58.56 | wc -l
50
```

  There are 50 connections that are open, which corresponds to the half of the 100 users that are defined in Swingbench.

6.4.5 Stopping Oracle on one node

In this phase, we complete the following steps to stop the Oracle cluster on ITSOTST6 and verify that the connections from Swingbench were rerouted to ITSOTST5:

1. Stop the Oracle cluster on ITSOTST6, as shown in the following example:

```
itsotst6:/home/grid # crsctl stop cluster
CRS-2673: Attempting to stop 'ora.crsd' on 'itsotst6'
CRS-2790: Starting shutdown of Cluster Ready Services-managed resources on
'itsotst6'
CRS-2673: Attempting to stop 'ora.DGOCRVOT.dg' on 'itsotst6'
CRS-2673: Attempting to stop 'ora.itsorac.db' on 'itsotst6'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on 'itsotst6'
CRS-2673: Attempting to stop 'ora.LISTENER_RDBMS.lsnr' on 'itsotst6'
…
CRS-2677: Stop of 'ora.cluster_interconnect.haip' on 'itsotst6' succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'itsotst6'
CRS-2677: Stop of 'ora.cssd' on 'itsotst6' succeeded
```

2. Check the status of the cluster during the process to stop the ITSOTST6 node, as shown in the following example:

```
grid@itsotst5:/home/grid> # crsctl status server
NAME=itsotst5
STATE=ONLINE

NAME=itsotst6
STATE=LEAVING
```

3. Recheck the status of the cluster after successfully stopping the ITSOTST6 node, as shown in the following example:

```
grid@itsotst5:/home/grid> # crsctl status server
NAME=itsotst5
STATE=ONLINE
```

4. Check the number of TCP sockets that are opened between Swingbench and ITSOTST5, as shown in the following example:

```
swingbench:~ # netstat -a | grep swingbench | grep 10.3.58.55 | wc -l
100
```

As shown in Figure 6-7, when the `ITSOTST6` is stopped, transaction per seconds froze for a short time on all the RAC. Some transactions are rolled back by the TAF and automatically balanced to the second node. However, all users stay connected to the database.

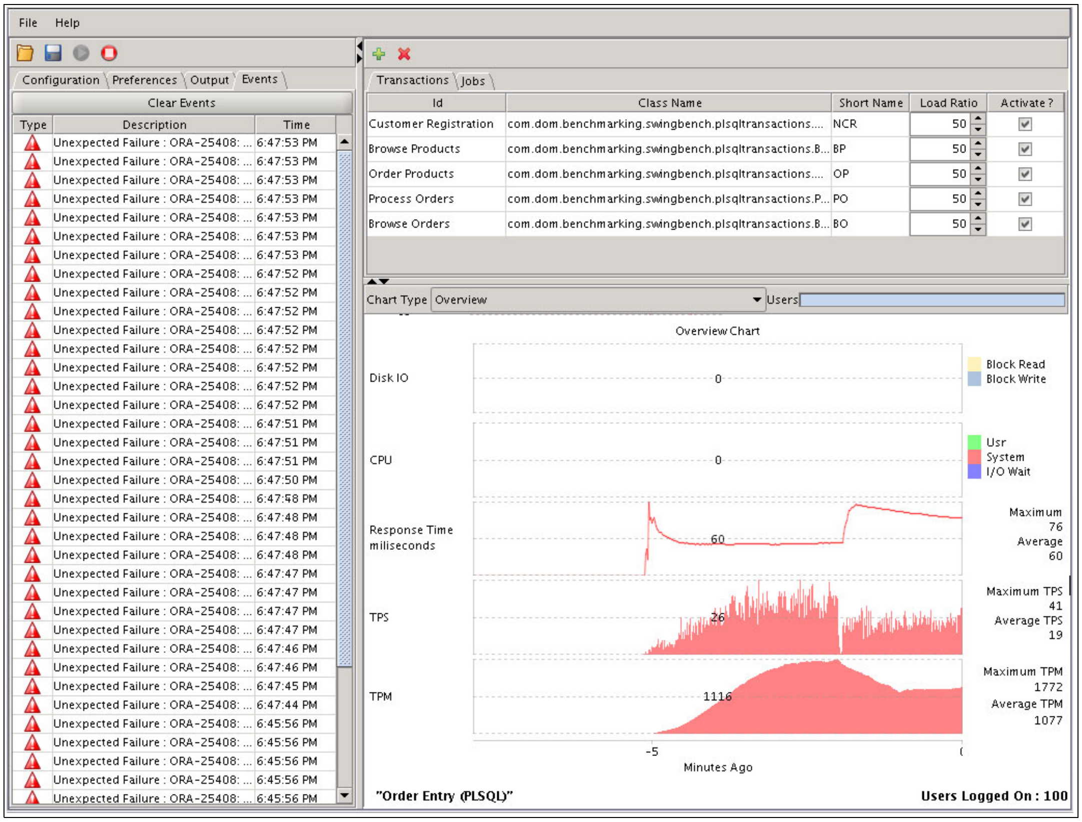


Figure 6-7 Swingbench GUI during relocation in Scenario 2

6.4.6 Relocating the Oracle guest

> **Class A user needed:** We run the z/VM commands from a class A user ID (for example, `MAINT`) of `MOPVMEM1`.

Complete the steps to perform guest relocation:

1. Check `ITSOTST5` is running in `MOPVMEM1`, as shown in the following example:

```
==> QUERY ITSOTST5 AT ALL
MOPVMEM1 : ITSOTST5 - DSC
```

2. Check `ITSOTST6` is running in `MOPVMEM1`, as shown in the following example:

```
==> QUERY ITSOTST6 AT ALL
MOPVMEM1 : ITSOTST6 - DSC
```

3. Test if the `ITSOTST6` can be relocated to `MOPVMEM2`, as shown in the following example:

```
==> VMRELOCATE TEST ITSOTST6 TO MOPVMEM2
User ITSOTST6 is eligible for relocation to MOPVMEM2
Ready; T=0.01/0.01 17:51:24
```

4. Relocate ITSOTST6 to MOPVMEM2, as shown in the following example:

```
==> VMRELOCATE MOVE ITSOTST6 TO MOPVMEM2
Relocation of ITSOTST6 from MOPVMEM1 to MOPVMEM2 started
User ITSOTST6 has been relocated from MOPVMEM1 to MOPVMEM2
Ready; T=0.01/0.01 17:52:31
```

5. Verify that the relocation was successful, as shown in the following example:

```
==> QUERY ITSOTST6 AT ALL
MOPVMEM2 : ITSOTST6 - DSC
```

6.4.7 Restarting Oracle on relocated guest

In this phase, we complete the following steps to restart the Oracle node on ITSOTST6 node and check the status of the cluster:

1. Start the Oracle node on ITSOTST6, as shown in the following example:

```
itsotst6:/home/grid # crsctl start cluster
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'itsotst6'
CRS-2676: Start of 'ora.cssdmonitor' on 'itsotst6' succeeded
CRS-2672: Attempting to start 'ora.cssd' on 'itsotst6'
CRS-2672: Attempting to start 'ora.diskmon' on 'itsotst6'
CRS-2676: Start of 'ora.diskmon' on 'itsotst6' succeeded
…
CRS-2676: Start of 'ora.asm' on 'itsotst6' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'itsotst6'
CRS-2676: Start of 'ora.crsd' on 'itsotst6' succeeded
```

2. Check the status of the cluster during Oracle start on ITSOTST6, as shown in the following example:

```
grid@itsotst5:/home/grid> # crsctl status server
NAME=itsotst5
STATE=ONLINE

NAME=itsotst6
STATE=JOINING
```

3. Recheck the status of the cluster after successful Oracle start on ITSOTST6, as shown in the following example:

```
grid@itsotst5:/home/grid> crsctl status server
NAME=itsotst5
STATE=ONLINE

NAME=itsotst6
STATE=ONLINE
```

After ITSOTST6 rejoins the cluster, the 100 users on ITSOTST5 cannot be rebalanced between the two nodes. Only new user connections are automatically balanced.

We can apply the steps that are described in Scenario 2 for Oracle cluster on `ITSOTST5`, relocate it with LGR, and then restart it.

By using Oracle failover capacities and z/VM LGR, we relocated all of our Oracle RAC infrastructure in a new z/VM with only a few seconds of interruption of services during the relocation and with no loss of users connections, queries, or transaction to the database.

7

Tuning z/VM, Linux, and Oracle to run on IBM System z

Attention: The following command conventions are used in this chapter:

- z/VM commands are prefixed with `==>`
- Linux commands that are running as root are prefixed with `#`
- Linux commands that are running as non-root are prefixed with `$`

This chapter describes performance tuning for Oracle on Linux on System z. Each subsystem (Oracle, Linux, and z/VM) is described.

Performance analysis is dependent on platform architecture. Performance analysis can be top down (z/VM to Linux to Oracle) or bottom up from the application perspective from Oracle to Linux to z/VM. In our case, we start with z/VM and Linux and then Oracle.

This chapter includes the following sections:

- Linux and z/VM
- Oracle
- Summary

7.1 Linux and z/VM

This section describes the aspects of Linux and z/VM performance tuning.

7.1.1 Architecture

In this section, the Linux and z/VM architecture is described.

Linux

From a Linux perspective, processor resource is assigned to kernel or processes. Processes can be Oracle processes, kernel processes, or from many infrastructure or other applications. The important part of analyzing the use of the processor at the Linux level is knowing which processes are using how much CPU. See Figure 7-4 on page 142 for an example.

When analyzing delay, it is common for installations to look at steal time. This is when the Linux server is waiting because another server or logical partition (LPAR) is using the CPU resource and Linux is not dispatched. This is the perfect example of the value of having higher level information: knowing there is a problem is not as valuable as knowing why there is a problem. From inside Linux, it is difficult to determine the total IFL usage that is causing the steal time.

z/VM

For our purpose, Linux servers run under z/VM, which, in turn, runs in one of possibly many LPARs. Given a specific number of IFL processors in one System z server, sharing of the processors is across LPARs, then virtual machines, and then within Linux might be across multiple virtual processors. The one metric that is important to monitor is the Total CPU Utilization. Processor distribution is handled within Linux by using nice, within z/VM by using share, and for LPARs with weights, none of which matter at low usage. At higher usage, it is up to the installation to decide which processes, virtual machines, and LPARs are allowed access to the real processors.

From a Linux perspective, nicing processes lower (negative) and gives them higher access to the processors. From a z/VM perspective, setting shares higher gives virtual machines more access to the physical processors. From an LPAR perspective, setting weights via the Hardware Management Console (HMC) determines which LPAR has access to how much processing power.

7.1.2 Best practices

In this section, the best practices for Linux and z/VM are described.

Linux

Best practices for Linux require an understanding of costs. This is a shared resource environment and resources that are used by untuned applications, unneeded applications, or infrastructure take away resources from productive work. An untuned application that uses excessive resources has a cost in terms of capital expenditures that are required to support that application because more IFLs are needed at some point than if the application was tuned. An IFL has a cost that can be identified.

z/VM

Without a full description of the z/VM scheduler, share settings are used to control CPU distribution, and can be absolute or relative. The following should be followed when you are setting share for virtual servers:

- There is a choice between absolute shares and relative shares. Both of these shares are normalized based on current load on the system in terms of logged on servers. The normalized share is used by the scheduler to schedule servers that are requesting service. Relative shares are provided to servers that offer a share of the processing power relative to other servers and that share drops as more servers log on to the system.

  Absolute shares are fixed and independent of load. As more servers log on to a system, they increase the server's share of the CPU resource as compared to relative servers. Thus servers where service requirements increase as load increases should be absolute. Servers that should compete for CPU resource should be relative. For example, most Linux servers should be relative; TCP/IP and RACF® should be absolute.
- Shares are divided by the number of virtual processors that are defined to the server. Thus, a default share of Relative 100 for a virtual server with two virtual processors provides both virtual processors a Relative share of 50. It is important to have a base line performance target (if there are two virtual processors) because the relative share should be 200. If there are four virtual processors, the relative share should be set to 400. Changing shares should be done in small increments.

7.1.3 CPU performance analysis

In this section, the CPU performance analysis for Linux and z/VM is described.

Linux

For more information about Linux performance on this architecture, see *Linux on IBM System z: Performance Measurement and Tuning,* SG24-6926.

z/VM

Performance analysis for the processor starts at the LPAR level. When there are multiple LPARs, as in Figure 7-1, the graph shows the total IFL utilization and the total General Purpose processor utilization.

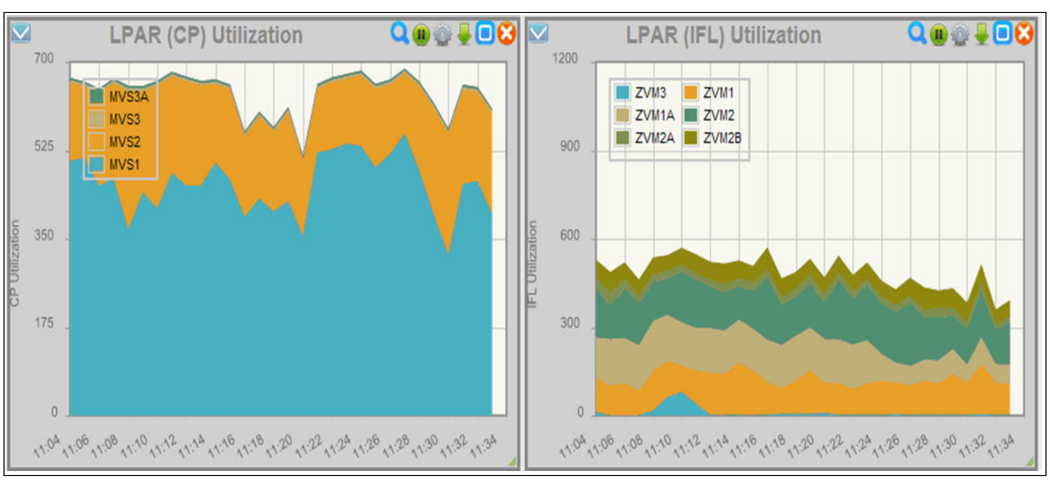


Figure 7-1 LPAR level view

7.1.4 Storage and memory performance

The storage and memory performance architecture for Linux and z/VM is described in this section.

Linux

Storage size is critical to Linux and Oracle performance. When the Oracle SGA/PGA does not fit into Linux storage, Linux storage management moves pages to the swapping subsystem. The following storage is measured:

- Kernel, including page structure tables
- Available, currently immediately usable by any Linux process
- Page Cache (where data and programs are) and the Oracle SGA
- Write Buffer, where Linux buffers output
- Anonymous storage, including short term working storage and page tables

Minimizing storage requirements in this shared storage environment means more storage is available for other servers and other work. In IBM System z environment, a much higher I/O bandwidth is supported and expected than distributed environments, which allows Linux on System z to reduce the cache size at the expense of more I/O. With the objective of maximizing throughput, oversizing storage reduces any configuration's work potential.

z/VM

The z/VM LPAR is allocated storage that is then shared between the virtual servers. When storage is over allocated and all virtual servers do not fit into storage, it is called *overcommit*, or *level of over commitment*. When storage is overcommitted, pages are paged to the paging subsystem on Expanded Storage or disk.

z/VM analysis

In environments where servers do not drop from the scheduler queue because of processes that poll on a 10 ms basis, it becomes difficult to differentiate between servers doing work and those that are only polling. When z/VM Control Program (CP) takes storage from a server that is polling, there is little cost if that page goes to disk. When CP takes a page from a server that is doing work, there is a high likelihood that the page is needed again by the server and there is a high cost when that page has gone to disk. Expanded Storage becomes the buffer, and the larger the buffer the better. In total, 20% - 25% of your storage should be configured as Expanded Storage to ensure active pages are not moved directly to disk.

> **Recommendation:** The 20% - 25% suggestion for Expanded Storage is for z/VM LPARs up to 8 GB. For larger z/VM LPARs, we suggest 2 GB - 4 GB of Expanded Storage, based on observed z/VM paging rates.

QDROP[1] is important. Avoid running "noisy" processes that poll every 10 ms, which keeps the server active. Servers that do not drop from queue require more real storage at a lower overcommit level.

Storage performance analysis

The storage performance analysis for Linux and z/VM is described in this section.

[1] QDROP is the process that CP uses to remove servers from the 'run queue', and thus defining them as inactive.

Linux

When storage is analyzed, the example in Figure 7-2 and Figure 7-3 shows three Oracle servers and their storage layout. Understanding from the Linux perspective the storage layout highlights the issues and the opportunities. In the following examples, the node `lnxsa3` is a 4 GB server with approximately 80 MB used for kernel and page structure tables (4096, subtract 4015.7). Swap space has few pages, so at one time this server used all of its storage, likely during the Oracle installation. The right buffer is 75 MB, and the page cache is 2,800 MB. Knowing the storage layout of all the servers shows where to focus from a system level.

```
Screen: ESAUCD2  ITSO                            ESAMON 4.110 10/17 13:55
1 of 2  LINUX UCD Memory Analysis Report         CLASS THEUSRS NODE * 282

                  <------------------Storage sizes in MegaBytes ------->
        Node/     <--Real Storage--> Over  <-----SWAP Storage----> Total
Time    Group     Total  Avail Used   head Total Avail Used  MIN   Avail
-------- -------- ------ ----- ----- ----- ----- ----- ----- ----- -----
13:56:00 lnxsa3   4015.7 892.9  3123 248.1 253.9 253.8   0.1  15.6  1147
         lnxsa1    996.5 502.6 493.9  94.9 253.9 219.0  34.9  15.6 721.6
         lnxcl2n1 4015.7  1222  2793 400.7  7296  7296     0  15.6  8519
```

Figure 7-2 LINUX UCD Memory Analysis Report, part 1

.

```
Screen: ESAUCD2  ITSO                            ESAMON 4.110 10/17 14:51-
2 of 2  LINUX UCD Memory Analysis Report         CLASS THEUSRS NODE * 2827

                  <------------------Storage sizes (in Megabytes) ------>
        Node/     <--Real Storage--> <--Storage in Use-> Error
Time    Group     Total  Avail Used  Shared Buffer Cache Message
-------- -------- ------ ----- ----- ------ ------ ----- ----------------
14:52:00 lnxsa3   4015.7 892.7  3123    0.0   75.2  2800
         lnxsa1    996.5 502.0 494.5    0.0   30.9 368.5
         lnxcl2n1 4015.7  1219  2796    0.0   40.7  2354
```

Figure 7-3 LINUX UCD Memory Analysis Report, part 2

To understand where the Linux storage is allocated, split screen mode is available, as shown in Figure 7-4 on page 142. In this case, the `LNXCL2N1` node was selected in split screen with the ESAUCD2 window showing the Linux perspective and the ESALNXP showing the RSS Resident Storage Size (RSS) for the active processes. The Linux storage management shares many of the pages between processes. Thus, the Oracle processes might each show 40 MB and 39 MB, the overlap between the two might be most of it. The “Shared” metric was not implemented in Linux now, thus it is not possible to know the overlap between these processes. In this case, the Java process is using considerably more than all the Oracle processes combined. This scenario highlights that because a server is Oracle, it should not create assumptions about how the storage resource is being used.

```
 Screen: ESALNXP  ITSO                           ESAMON 4.110 10/18 13:54-13:55
 2 of 3  VSI Linux Percent Usage by Process      NODE LNXCL2N1 DEVICE 2827

                           <-Process Ident-> <--Memory-> <--Faults/Second-->
 Time     Node     Name        ID  PPID   GRP  Size   RSS  min  maj mint majt
 -------- -------- --------- ----- ----- ----- ----- ----- ---- ---- ---- ----
 13:52:00 lnxcl2n1 *Totals*      0     0     0   69G 1901M   98    0  785    0
                   init          1     1     1 2538K  893K    0    0  301    0
                   snmpd      2610     1  2609   27M 9744K   21    0    0    0
                   ohasd.bi   2688     1  2688  222M   40M    0    0    0    0
                   oraagent   3039     1  3039  269M   39M    7    0   49    0
                   oracle     3207     1  3207  479M   15M    0    0    0    0
                   oracle     5126     1  5126 1925M   17M    0    0    0    0
                   java      26947 26902 26902  453M  162M   68    0  429    0


PF1=Help                PF3=Quit  PF4=Select  PF5=Plot   PF6=ESALNXR PA1=CP
PF7=Backward  PF8=Forward           PF10=Parms  PF11=More  PF12=Exit
PA2=Copy
====>
 Screen: ESAUCD2  ITSO                           ESAMON 4.110 10/18 13:54-13:55
 2 of 2  LINUX UCD Memory Analysis Report        CLASS THEUSRS NODE L 2827

                   <------------------Storage sizes (in Megabytes) ------>
          Node/    <--Real Storage--> <--Storage in Use-> Error
 Time     Group    Total  Avail Used  Shared Buffer Cache Message
 -------- -------- ------ ----- ----- ------ ------ ----- ----------------
 13:55:00 lnxcl2n1 4015.7  79.4  3936    0.0  144.2  3167


PF1=Help      PF2=Zoom     PF3=Quit  PF4=Select  PF5=Plot              PA1=CP
PF7=Backward  PF8=Forward            PF10=Parms  PF11=More   PF12=Exit
PA2=Copy
```

Figure 7-4 Split Screen Mode showing Linux storage and active processes

z/VM

Real storage in a z/VM system is assigned to one of many different types of address spaces. To analyze storage, first understand where the existing storage is allocated. The example in Figure 7-5 shows from the system level that there are 6.8 million pages.

```
Screen: ESASTR1  ITSO                           ESAMON 4.11
1 of 2  Main Storage Analysis

         <-----------------Pages-----------------> Capt-
         System  Fixed Non-  Free Frame <Available >  ure
Time     Storage Store Pgble Stor Table <2gb  >2gb   Ratio
-------- ------- ----- ----- ---- ----- ----- ------ -----
12:06:00 6815744  2458  5058  252 53248  178K  2094K 0.998
12:05:00 6815744  2458  5058  252 53248  178K  2094K 0.998
```

Figure 7-5 Main Storage Analysis

In Figure 7-6, storage for each address space type is found. In this case, users are using 2.1 million pages, saved segments are using 130 K pages, VDisk address spaces are using 56,000 pages, and minidisk cache (MDC) has 175K pages. MDC should never be allowed to use that much storage, especially in a Linux environment. The exception is for shared root and binary implementations where multiple servers are sharing some number of minidisks read only.

```
Screen: ESASTR1  ITSO                            ESAMON 4.110 10/18
2 of 2  Main Storage Analysis

         <----------------------Pages----------------------> Capt-
         Systm  User  NSS/DCSS  <-AddSpace> VDISK <MDC> Diag   ure
Time     ExSpc Resdnt Resident  Systm User  Rsdnt Rsdnt  98  Ratio
-------- ----- ------ --------  ----- ----- ----- ----- ---- -----
12:55:00  5148  2105K   139189  47416     0 56597  175K 2304 0.998
12:54:00  5142  2106K   139186  47416     0 56597  175K 2304 0.998
12:53:00  5145  2107K   139186  47416     0 56597  175K 2304 0.998
```

Figure 7-6 Main Storage Analysis page 2

Each type of address space can be further analyzed. For example, the number of pages in use by Named Saved Systems (NSS) and Discontiguous Saved Segment (DCSS) seemed high at 139,189 pages. So, in looking at the ESADCSS display (as shown in Figure 7-7), the `VSMDCSS` is more than 90% of that and `VSMDCSS` was the SMAPI saved segment.

```
Screen: ESADCSS  ITSO                            ESAMON 4.110
1 of 3  NSS/DCSS Analysis                        DCSS *

                              <----------Number of Pages for DCSS-
                  Spool       Non- <--Resident--> <-Locked>
Time     Name        ID Saved Data <2GB >2GB HOST <2GB >2GB
-------- -------- ----- ----- ---- ---- ---- ---- ---- ----
13:14:00 System       0  6282 165K  12K 127K  369    0  105
         CMS         44  1298    0    0 1288  257    0    0
         CMSFILES    10   768    0    0  635    0    0    0
         CMSPIPES     8   256    0    0  256    0    0    0
         CMSVMLIB     7   256    0    0  249    0    0    0
         GCS          3   120  917   98  939    0    0    0
         INSTSEG      6   768    0    4  635    0    0    0
         MONDCSS     12     0  16K    0  111  111    0  105
         NLSAMENG    38   256    0    0  128    0    0    0
         SCEE        13   256    0    0  128    0    0    0
         SCEEX       14  2304    0    0 2044    0    0    0
         VSMDCSS     47     0 131K  10K 121K    0    0    0
         ZMON        42     0  16K 1553    0    1    0    0
         ZVWS        43     0  256    0   55    0    0    0
```

Figure 7-7 NSS/DCSS Analysis

A best practice for managing many Linux servers includes classifying those servers. In Figure 7-8, our Cloned Linux servers are in `TheUsrs`, with other servers classified per their use. In this analysis, because we were concerned about how much storage was going to be used when we started 100 servers, it was important to watch how much storage was used to avoid having the whole system abend because of a lack of resources (our paging subsystem was only 8 GB).

```
Screen: ESAUSPG  ITSO                           ESAMON 4.110 10/18 15:47-15:4
1 of 2  User Storage Analysis                   CLASS *             2827 4B8

                 <---Storage occupancy in pages--->  Pages <Address Spaces>
        UserID   <---Main Storage---> <--Paging--->  Moved <Pages Resident>
Time    /Class    Total  >2GB   <2GB   Xstor  DASD    <2GB VirtDisk AddSpce
-------- -------- ------ ------ ------ ------ ------ ------ -------- -------
15:48:00 System:   6112K  5627K 484943  20512     41      0   132629       0
         ORACloud  1785K  1638K 146861    108     40      0    73520       0
         Cluster1  1527K  1408K 119215  10460      0      0    31154       0
         Cluster2  1463K  1349K 114295    422      0      0     9369       0
         StnAlone  1197K  1102K  94453      0      0      0    16721       0
         OthLinux  60550  55613   4937      0      1      0     1838       0
         TheUsrs   42611  39452   3159   1657      0      0        0       0
         Servers   21280  20739    541   5278      0      0        0       0
         Velocity  11806  11085    721    745      0      0       27       0
         KeyUser    4323   3562    761   1842      0      0        0       0
```

Figure 7-8 User Storage Analysis

With a potential for 100s of servers on the LPAR, it is more important to understand resource requirements by workload rather than by server. After the workload is analyzed, in this case (as shown in Figure 7-9), the next step was to look at the cloud workload. By carefully reviewing the `ORACLOUD` workload, all the servers in that workload show up. In this experiment, we cloned 100 servers in about 60 minutes and then logged them on.

```
Screen: ESAUSPG  ITSO                           ESAMON 4.110 10/18 15:50-15:5
1 of 2  User Storage Analysis                   CLASS ORACLOUD USER  2827 4B8

                 <---Storage occupancy in pages--->  Pages <Address Spaces>
        UserID   <---Main Storage---> <--Paging--->  Moved <Pages Resident>
Time    /Class    Total  >2GB   <2GB   Xstor  DASD    <2GB VirtDisk AddSpce
-------- -------- ------ ------ ------ ------ ------ ------ -------- -------
15:51:00 S11OR001  51071  46895   4176      0      1      0     1838       0
         S11OR003  49872  45831   4041      0      1      0     1838       0
         S11OR002  49794  45718   4076      0      1      0     1838       0
         S11OR025  49602  45543   4059      0      1      0     1838       0
         S11OR039  49542  45185   4357      0      1      0     1838       0
         S11OR040  49541  45729   3812      0      1      0     1838       0
         S11OR022  49518  45243   4275      0      1      0     1838       0
         S11OR030  49512  45337   4175      0      1      0     1838       0
         S11OR024  49506  45204   4302      0      1      0     1838       0
         S11OR032  49506  45590   3916      0      1      0     1838       0
         S11OR013  49423  45561   3862      0      1      0     1838       0
         S11OR018  49401  45508   3893      0      1      0     1838       0
```

Figure 7-9 User Storage Analysis for ORACLOUD

7.1.5 Disk I/O performance

DASD performance is critical to a database application. The following technology questions can be considered:

- Extended count key data (ECKD) versus Fibre Channel Protocol (FCP)
- Disk size or logical unit number (LUN) size
- HiperPAV or parallel access volume (PAV)
- Ficon Channel Extension (FCX)/High Performance Ficon (HPF)
- Logical Volume Manager (LVM), striped or non-striped and strip size
- Linux I/O Scheduler
- DIRECT I/O versus buffered

Different technologies have different costs, performance, and management capabilities. For more information about disk I/O and Oracle, see the *Oracle Database on Linux on System z – Disk I/O Connectivity Study*[2] white paper. It provides guidelines for disk configuration and tuning hints regarding ECKD and FCP disk devices, specifically for the Oracle stand-alone database that uses a transactional workload (OLTP) in an LPAR on IBM zEnterprise System.

ECKD versus FCP

The following primary considerations should be reviewed when you are choosing the disk technology:

- Size of the database: For large databases, FCP is the most reasonable in terms of cost and manageability. When database sizes are measured in terabytes or petabytes, it is better to allocate disk space in terms of large LUNs on FCP.
- Performance requirements: When the ability to manage performance is required, data should be on ECKD because of its extensive performance reporting capability. This includes ALL system devices and binaries.

Overall, FCP offers better throughput and performance, but ECKD uses less CPU per transaction.

Disk size or LUN size

The size of the allocated unit has performance implications. For traditional 3390 DASD with a choice of 3390-3, 3390-9, or 3390-27, the question is bandwidth. If higher bandwidth in terms of I/O per second is required, the choice is smaller devices and more of them supporting higher I/O rates, or the use of PAV or HyperPAV.

PAV is an older technology that allows multiple device exposures to be defined for one real device, which increases the number of concurrent I/O to the “base” exposure.

HyperPAV is a newer and more dynamic technology that allows alias devices to be configured as needed for base devices when base devices require concurrent I/O.

z/VM disk considerations

FCX or HPF are two names for the same feature. This feature increases data transmission rates on Ficon channels for the software that supports it.

ECKD is recommended for all CP system devices and other data characterized as important for performance. Ensure there are enough devices so that device usage is less than 20% device busy.

[2] http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102234

7.2 Oracle

This section describes the Oracle specific aspects of performance tuning.

7.2.1 Architecture

Oracle is a standard Linux application, in that it uses various processes to perform tasks on behalf of the database and its users. Each database process allocates the Oracle kernel and runs the code to address the System Global Area (SGA). For this reason, the SGA must be in a shared memory area. It is this method that permits each process to believe that the SGA is uniquely assigned to it, and yet shares it among other processes.

Oracle processes consist of the following tasks:

- Database background tasks: These tasks perform session management and cleanup, write I/O, cache management, pool management, and lock detection. They also manage logging, log archival locally and remotely, check pointing, and corruption detection. In a clustered environment, they manage inter-instance communication, log management, and cluster membership. In an Automatic Storage Management (ASM) configuration, they manage segment allocation and storage pool management. Other tasks include manage time, job scheduling, memory management, and Automatic Workload Repository (AWR) statistics collection.
- Database server processes: User tasks that perform operations on behalf of user requests. They can exist one per user session (Dedicated Server), or as Shared Servers that can be used to multiplex user sessions to processes, which saves resources.

User processes are critical for performance measurement because they run the SQL and also perform read I/O on behalf of the user session into the buffer cache. Because a database is principally designed to service SQL and perform I/O to get the data to perform that service, most performance measurement and management is concentrated on these processes.

For more information about the Oracle Process architecture, see Chapter 15 *Process Architecture of Oracle Database Concepts 11g Release 2 (11.2)*, E25789-01, which is available at this website:

http://docs.oracle.com/cd/E11882_01/server.112/e25789/process.htm#i7265

7.2.2 Best practices: CPU

It is important to ensure that the operating system has sufficient resources to run the required number of background processes; for example, the following installation guide shows:

- oracle: soft, nproc, 2047
- oracle: hard, nproc, 16384

However, these need to be adjusted for large systems and user populations.

In addition, the `init.ora processes` parameter should be set higher than the expected user population to allow for growth and the background processes, which are frequently overlooked, such as parallel processes and background jobs. While the `processes` parameter does contribute to some storage settings internally, in modern 64-bit systems, such as, Linux on System z, the overhead of setting this parameter too high is trivial.

It is critical to keep up with maintenance and patching. For example, with Release 11.2.0.3 the following parameters exist:

- Virtual Keeper of Time (VKTM) process uses slightly less CPU minutes (about 0.08 versus 0.09 with 11.2.0.2).
- Great improvements with `ora_dia0` process (about 0.07 sec cpu/minute versus 0.28 with 11.2.0.2).

Install only the database modules that are needed and be aware of the following results:

- When the database is installed with no options, the `gettimeofday` function is called 300 times every 15 seconds.
- When the database is installed with all options (Java, `.xml`, Text, spatial, APEX, and so on), the `gettimeofday` function is called 1500 times every 15 seconds.

Consider whether it is appropriate to use the Oracle Resource Manager, especially under z/VM because you then have multiple layers of CPU scheduling and slicing potentially interfering. You can disable this with the following Oracle parameter:

```
resource_manager_plan = ''
```

Additionally, you need to disable the Maintenance Window Resource Plan, as shown in the following example:

```
select window_name,RESOURCE_PLAN from DBA_SCHEDULER_WINDOWS;
WINDOW_NAME RESOURCE_PLAN
------------------------------ ------------------------------
MONDAY_WINDOW DEFAULT_MAINTENANCE_PLAN
execute dbms_scheduler.set_attribute('MONDAY_WINDOW','RESOURCE_PLAN','');
WINDOW_NAME RESOURCE_PLAN
------------------------------ ------------------------------
MONDAY_WINDOW
```

7.2.3 Best practices: Performance analysis

Two standard methods of reviewing performance from the database are available: *statspack* and *AWR*. Statspack is a non-chargable add-on. AWR is part of the licensable Diagnostic and Tuning pack. When used with `sysstat`, these methods provide the basis for performance measurement for the Oracle Database.

AWR has many significant advantages in that it can report at an Active Session level, on a snapshot interval, and, importantly, at a cluster level for RAC clusters.

From the process perspective, it is important that processes receive the resources that are required to perform their task because the peak CPU demand can be 100% but the CPU delay should not affect performance. The simplest measure for this is the following equation:

```
SQL Elapsed time = CPU Time + Wait time
```

If the Wait Time is composed of active waits (I/O, for example), and the per block I/O time is in the normal range for the system, the process that is running the SQL is not CPU-starved.

In a similar manner, we can check whether the process in I/O bound and take appropriate action.

For more information about using AWR to diagnose performance issues, see *How to Use AWR reports to Diagnose Database Performance Issues*, 1359094.1.

Based on a real-world scenario of a RAC system, we show a worked example in Figure 7-10.

WORKLOAD REPOSITORY REPORT (RAC)

Database Summary

Database				Snapshot Ids		Number of Instances		Number of Hosts		Report Total (minutes)	
Id	Name	RAC	Block Size	Begin	End	In Report	Total	In Report	Total	DB time	Elapsed time
3614964357	OASTDB1	YES	8192	447	449	4	4	4	4	6,035.04	65.13

Database Instances Included In Report

- Listed in order of instance number, I#

I#	Instance	Host	Startup	Begin Snap Time	End Snap Time	Release	Elapsed Time(min)	DB time(min)	Up Time(hrs)	A
1	oastdb11	.com	08-Oct-12 10:04	08-Oct-12 10:12	08-Oct-12 11:17	11.2.0.3.0	65.05	842.88	1.21	
2	oastdb12	.com	08-Oct-12 10:07	08-Oct-12 10:12	08-Oct-12 11:17	11.2.0.3.0	65.07	1,717.72	1.17	
3	oastdb13	.com	08-Oct-12 10:07	08-Oct-12 10:12	08-Oct-12 11:17	11.2.0.3.0	65.05	1,288.48	1.17	
4	oastdb14	.com	08-Oct-12 10:07	08-Oct-12 10:12	08-Oct-12 11:17	11.2.0.3.0	65.13	2,185.97	1.17	

Figure 7-10 AWR example

Here, the DB Time is a high multiple of the elapsed time, which indicates that this system and all nodes on the cluster are busy.

If we look at the top timed events that are shown in Figure 7-11 (remembering that these are not necessarily waits), we see that most of the time is spent in I/O and RAC interconnect. However, what is important to notice is that the average response times for log file sync are 245 ms, interconnect 30 ms - 200 ms, and even local I/O 42 ms. These numbers are unusual for a System z, and we can suspect that what we see from the AWR is an artifact of another problem.

Top Timed Events

	Wait		Event		Wait Time			Summary Avg Wait Time (ms)				
I#	Class	Event	Waits	%Timeouts	Total(s)	Avg(ms)	%DB time	Avg	Min	Max	Std Dev	Cnt
*	Commit	log file sync	495,221	0.00	121,367.27	245.08	33.52	232.27	145.94	310.55	76.94	4
	User I/O	db file sequential read	1,881,165	0.00	79,121.56	42.06	21.85	44.02	35.45	60.77	11.43	4
	Cluster	gc cr grant 2-way	838,384	0.00	25,494.35	30.41	7.04	30.53	21.87	39.37	8.68	4
	Cluster	gc current grant 2-way	878,191	0.00	21,529.29	24.52	5.95	24.54	16.94	32.81	7.11	4
	Cluster	gc cr block 3-way	300,081	0.00	9,249.76	30.82	2.55	30.99	21.15	44.03	10.73	4
	Cluster	gc cr grant congested	42,721	0.00	8,938.08	209.22	2.47	207.16	168.53	244.44	39.93	4
	Cluster	gc current grant congested	43,524	0.00	7,078.83	162.64	1.95	161.49	124.52	203.71	35.72	4
	Cluster	gc cr block 2-way	326,543	0.00	5,843.02	17.89	1.61	18.45	12.04	25.43	7.00	4
		DB CPU			5,575.75		1.54					4
	Cluster	gc current block 3-way	191,226	0.00	5,530.42	28.92	1.53	28.89	19.95	39.17	8.90	4

Figure 7-11 Top Timed Events

Review the OS statistics that are shown in Figure 7-12.

OS Statistics By Instance

- Listed in order of instance number, I#
- End values are diplayed only if different from begin values

I#	Num CPUs	CPU Cores	CPU Sckts	Load Begin	Load End	% Busy	% Usr	% Sys	% WIO	% Idl	Busy Time (s)	Idle Time (s)	Total Time (s)	Memory (M)
1	2	1	1	0.53	30.25	81.53	62.49	14.10	13.01	18.47	5,408.65	1,225.32	6,633.97	10,043.69
2	2	1	1	0.15	30.38	83.66	56.40	21.65	11.84	16.34	5,380.47	1,050.57	6,431.04	10,043.69
3	2	1	1	0.33	41.31	75.06	57.77	12.43	13.25	24.94	5,063.40	1,682.15	6,745.55	10,043.69
4	2	1	1	0.21	36.70	79.89	61.21	13.39	8.89	20.11	5,351.72	1,347.11	6,698.83	10,043.69
Sum											21,204.24	5,305.15	26,509.39	

Figure 7-12 OS Statistics By Instance

We are in general, over 75% are busy (as an average over the period), which does not seem too bad. However, AWR is not virtual aware and Oracle is unaware that it is not running on a dedicated machine. By taking a step back and looking through the Linux performance view, perhaps we can see the cause of the issue.

In this case, we use the **vmstat** command to see the Linux view of performance, as shown in Figure 7-13.

procs ---		--------memory---------				---swap--		-----io----		--system--		-----cpu-----				
r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st
233	12	11280	3047908	178448	4755356	0	0	2317	693	5673	9337	34	13	31	5	17
3	75	11280	2757980	178752	4765204	0	0	45246	638	4501	8136	31	10	4	37	17
74	9	11280	2741332	179040	4767624	0	0	133886	464	4971	8566	58	12	0	8	23
51	3	11280	2728628	179340	4770144	0	0	119453	529	6719	11139	77	14	0	0	9
18	18	11280	2592464	179644	4947080	0	0	140365	1140	7375	11744	68	19	1	1	11
40	12	11280	2297332	179936	5238692	0	0	131479	1164	7272	11857	68	17	1	2	12
123	2	11280	274212	180816	7201132	0	0	40815	656	4100	7134	36	12	11	11	30
25	12	50200	44536	181104	7273384	0	649	17410	1165	3399	5768	25	15	4	19	37
34	2	62572	95844	179464	7141368	0	1875	11510	2383	3673	6568	17	10	0	52	21
143	4	269084	70248	178032	7040720	0	1779	16294	2391	4373	7652	28	12	0	38	23
55	9	371752	46732	160232	6991416	1	1741	22691	3018	5205	8277	34	20	0	23	23
155	12	335344	93184	117164	7021612	105	451	32133	1753	5655	9386	46	19	1	9	25
34	4	335748	78816	108036	7016056	0	20	50505	1423	6453	10747	53	16	3	14	14
12	14	336844	46664	59888	7033256	1	48	63910	1260	5888	9489	57	21	0	5	17
16	15	335640	79812	52924	7073776	2	4	74431	1583	7269	11575	54	19	0	14	12
7	19	336048	97296	30016	7008656	0	17	155967	841	6230	10617	55	20	0	10	15
32	11	335584	118624	30096	6999412	0	5	156307	1229	6917	11284	56	17	0	12	14
67	37	335132	72620	30208	7030260	0	10	164980	1472	6967	11265	58	20	0	12	10
28	11	406052	164488	7576	6968076	1	190	99400	2459	7035	11228	49	21	0	19	11
20	1	428672	60772	7340	6966576	1	390	103906	1991	7259	11421	55	21	0	13	12

Figure 7-13 VMstat output

We can readily see that we have high run queue, which indicates that we cannot service the resource demand. We have rapidly reducing free memory, and we can see that although AWR reports free CPU, we can see that this is taken up by steal time, which AWR does not take into account.

To summarize this example, the guests were running with insufficient memory to service the SGA and pGA demands, which led to swapping. The CPU demands of this swapping across the LPAR reduced the CPU available to service the database, which leads to a process queue building up of both database tasks, network tasks, and I/O tasks, which are reported in the AWR as slow I/O and interconnect. This is an example of how a shortage of one resource can appear as a different issue and is only diagnosable by using multiple layers of diagnostic information.

For more information about Linux performance on this architecture, see *Linux on IBM System z: Performance Measurement and Tuning,* SG24-6926.

7.2.4 Best practices: Storage and memory

When memory is allocated for Oracle, it is important to understand that the primary allocation is to the SGA, and that this memory, while eligible for swap, should not be swapped if possible. The reason for this is that Oracle, Linux, and z/VM all manage memory on a Least Recently Used (LRU) basis, which means that the SGA page Linux deems most eligible for swap is the page Oracle deems most eligible for reuse. Add in a virtual memory manager in the hypervisor (VM) and it is easy to see why some systems spend most of their time thrashing in memory allocation (which means CPU usage) rather than working effectively. For this reason, it is recommend that large or huge pages are used to fix the SGA in memory, which reduces the UNIX overheads of large memory allocations and ensures that the process access to the SGA is as efficient as possible.

The Oracle Database 11g AMM feature is enabled by the MEMORY_TARGET / MEMORY_MAX_TARGET instance initialization parameters. These cause the SGA to be allocated in the /dev/shm temporary space, and this must be configured to be as large as the sum of MEMORY_TARGET / MEMORY_MAX_TARGET for all instances that use AMM on the guest or LPAR.

One of the principal issues with large SGA sizes with large user populations is the amount of memory that is required to manage the memory structures: the page tables. Each Oracle process must map the SGA as though it were a private memory area and maintain a page table to perform the page translation from the process local address to the real page address in the shared memory segment. These tables take 4 bytes per 4k SGA page addressed, so it can easily be seen that a 60 GB SGA with 3500 active users needs an enormous amount of memory to maintain these tables. This memory often is not tracked, and if you are not aware of it, it can cause some unexpected surprises and out-of-memory conditions. To view the usage, you can use the following command:

```
$ cat /proc/meminfo | grep page
AnonPages:          1039248 kB
PageTables:         98820932 kB
HugePages_Total:          0
HugePages_Free:           0
HugePages_Rsvd:           0
Hugepagesize:       2048 kB
```

The best way to avoid this condition is to use Hugepages to increase the page size of the shared memory segment, which reduces the number of mapping page tables that are required per process.

Using Hugepages with Oracle Shared Memory Management

The Oracle Hugepage feature is enabled by using the Oracle Shared Memory Management (ASMM) parameters SGA_TARGET / SGA_MAX_TARGET.

You can also enable Hugepages by setting the Oracle manual SGA memory instance parameters (shared_pool_size, db_cache_size, large_pool_size, java_pool_size, and streams_pool_size).

Only the Oracle SGA is eligible for Hugepages. Sufficient remaining memory for pGA memory (PGA_AGGREGATE_TARGET) should be accounted for when sizing the number of Hugepages for the database.

Note: The Oracle Automatic Memory Management parameters MEMORY_TARGET and MEMORY_MAX_TARGET are currently incompatible with Hugepages, so they cannot be used.

To configure Hugepages, you first must define the number of kernel Hugepages, which is set by using the kernel parameter, as shown in the following example:

```
vm.nr_hugepages = n,
```

Where n is the sum of the SGAs that are planning to use Hugepages (in MB) divided by the Hugepages size (from /proc/meminfo - 1 MB on System z).

Use the following note to calculate this value for you: *Shell Script to Calculate Values Recommended Linux HugePages / HugeTLB Configuration*, Doc ID 401749.1, which is available at this website:

https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=401749.1

In either case, you must also increase the `ulimit` parameter memlock in `/etc/security/limits.conf`.

Set the value (in KiB) slightly smaller than installed RAM. For example, if you have 64 GB RAM installed, you can set the following parameters:

```
*   soft   memlock    60397977
*   hard   memlock    60397977
```

We also strongly recommend setting the Oracle parameter as shown in the following example:

```
use_large_pages = ONLY
```

This ensures that you do not use small pages when you expect to use large or Hugepages. The implications of this on a busy production system can be serious performance issues.

Set `ONLY` for those instances where you expect hugepages, and `FALSE` for those that you know do not require the small SGA, in general.

> **Tip:** For more information about Hugepages, see the following My Oracle Support Notes:
>
> - *HugePages on Linux: What It Is... and What It Is Not...*, 361323.1
> - *HugePages and Oracle Database 11g Automatic Memory Management (AMM) on Linux*, 749851.1
> - *ORA-00845 When Starting Up An 11g Instance With AMM Configured*, 460506.1
> - *USE_LARGE_PAGES To Enable HugePages In 11.2*, 1392497.1

7.2.5 Best practices: I/O

Use Direct and Asynchronous I/O by setting the Oracle parameters as shown in the following example:

```
filesystemio_options = SETALL
disk_asynch_io = TRUE
```

ASM forces Direct I/O, but uses the database async setting at file open time. It is important to understand that ASM does not perform I/O, it simply manages the disk and the segment storage on those disks, so many elements of database I/O performance tuning are still the same; for example, ASM has no buffer cache.

Direct I/O bypasses the Linux disk cache, so only a small area is required for this to permit frequently used objects, principally executable files and non Oracle buffer cache managed files to be buffered.

Kernel I/O scheduler

The Linux kernel offers a choice of the following I/O schedulers:

- Noop Scheduler (Noop)
- Deadline Scheduler (Deadline)
- Complete Fair Queuing Scheduler (CFQ)
- Anticipatory Scheduler (as)[3]

The Linux default is the CFQ scheduler for 2.6 Linux kernels and deadline for 3.x kernels.

[3] No Longer available in Red Hat 3.x Kernels

CFQ is optimized for access to direct physical disks and is not suitable for typical storage servers that are used with System z.

The I/O scheduler is configurable by setting the `elevator=` boot parameter in the `/etc/zipl.conf` file.

You can verify your current settings by checking the following parameters:

```
# cat  /sys/block/<device>/queue/scheduler    ( [... ] is current setting)
noop  anticipatory  [deadline]  cfq
```

For Linux on System z Oracle environments, we recommend the Deadline or Noop I/O scheduler.

It is important to investigate the best I/O scheduler for your environment by testing with tools such as Oracle's io_calibrate or Orion.

On one system, we changed the I/O scheduler in the `zipl.conf` parameters `elevator=`**`noop`**, which helped with reducing CPU for the SAN environment that we were using.

If a non-ASM file system is used for your database files, it is recommended to reduce the Linux ReadAhead for your database LVM file systems with the following command:

```
# lvchange -r none <lv device name>
```

File system options

The following file system options are available:

- ext2: Most widespread Linux file system
- ext3: Evolved from ext2 and adds journaling features
- ext4: Only supported by Red Hat Enterprise Linux 5.6, Oracle Linux 5.6, SUSE Linux Enterprise Server 11, or greater (recommend testing first)
- jfs: A port of OS/2 Warp Server jfs to Linux
- reiserfs: Journaling behavior is comparable to ext3 in order mode

It is recommended that ext3 or ext4 is used because of its journaling capabilities and reduced CPU load. For more information, see *Performance of Large Journaling File Systems*, which is available at this website:

http://download.boulder.ibm.com/ibmdl/pub/software/dw/linux390/perf/ZSW03027-USEN-00.pdf

Many of the established sets of Performance Management principles were based on having to cater for the following older technologies:

- Slow disk: 20 ms service time
- Slow networks
- Intermittent peaks
- Slow provisioning or long lead time

The following changes were made:

- Virtualization: Rapid provisioning and resource sharing
- Faster disk: 0 ms - 10 ms
- Faster networks: Gigabit is normal
- Need to use 100% capacity

It used to be said that, Because disk I/O is slow, make use of large Oracle and OS buffers. Now, with much faster disk (especially in the low ms service time range), disk is still not as fast as memory but it is possible for random transactional I/O to use disk. If a transaction can get its data in a few milliseconds, overall transaction time is unaffected.

> **Tip:** Use direct, asynchronous I/O to bypass the OS cache and tune the Oracle buffer cache for optimal operation when you are migrating from another platform because this usually means reduce extensively.

It used to be said that you should avoid network operations. While this is still true, SAN devices over modern channel and network attachments coupled with multipath or PAV means that network or pseudo network devices are no longer the slow option they used to be. Drive the I/O subsystem as hard as you can. You paid for the capacity, so use it.

7.3 Summary

This is a constantly evolving subject. The latest information about this topic often is presented at the Oracle Collaborate, SHARE User Group, and zSeries® Oracle Special Interest Group Conferences.

For more information, see this website:

http://www.zseriesoraclesig.org

8

Cross-platform migration overview

Attention: The following command conventions are used in this chapter:
- Linux commands that are running as root are prefixed with #
- Linux commands that are running as non-root are prefixed with $

In this chapter, we provide an overview of cross-platform migration topics.

We first highlight what you need to consider before any migration to choose the appropriate technique. Then, we describe the most popular migration techniques with a set of best practices to perform an efficient migration.

We also provide a real example in which we describe the steps of a migration by using Oracle Export/Import Data Pump utilities.

This chapter includes the following topics:
- Introduction
- Considerations before any migration
- Migration techniques
- Best practices
- Example of migration by using Export/Import Data Pump
- Summary

8.1 Introduction

Cross platform migration became a common operation in the IT world. Companies perform this operation for the following reasons:

- IT optimization and consolidation
- Companies merging
- End of life servers
- Software support modifications (for example, Oracle support on HP Itanium)
- Other platform changes

After the decision to migrate from one platform to another is made, several options are available. The choice of the migration technique is guided by technical criteria and organizational criteria.

In this chapter, we describe the most used techniques (based on our experience) to give an overview on what is available and the advantages and the limitations of each technique. We do not recommend one technique more than another because each case is specific. Furthermore, migration can also combine several techniques.

In our example, we focus on the Oracle Export/Import Data Pump utility, which is likely the most used technique for migrating from one platform to another.

This document does not replace any IBM or Oracle documents. We assume that the user is reasonably skilled in the following areas:

- Oracle database administration activities
- Linux System administration skills

8.2 Considerations before any migration

The choice of the migration technique depends on client requirements, technical criteria, and organizational criteria.

8.2.1 Downtime

Each migration leads to a database downtime. Depending on the technique, this downtime can be from a minimum of a few minutes to more than a day. For critical applications that must be always available, downtime is the main criteria for clients to choose the appropriate technique.

8.2.2 Technical compatibility

From a technical point of view, you should be aware of some limitations.

Endianness

The Endianness describes how the bits are organized as seen from the outside. Depending on the platform, this can be “Little Endian” or “Big Endian”.

Some cross platform migration methods require the same Endianness.

The following queries can be used to determine the Endianness:

- Use the following command to check the Endianness for all platforms:

```
select * from v$transportable_platform;
```

  The command output is shown in Table 8-1.

Table 8-1 Endianness by platform

PLATFORM_NAME	ENDIAN_FORMAT
Solaris™ OE (32-bit)	Big
Solaris™ OE (64-bit)	Big
Microsoft Windows IA® (32-bit)	Little
Linux IA (32-bit)	Little
AIX-Based Systems (64-bit)	Big
HP-UX (64-bit)	Big
HP Tru64 UNIX	Little
HP-UX IA (64-bit)	Big
Linux IA (64-bit)	Little
HP Open VMS	Little
Microsoft Windows IA (64-bit)	Little
IBM zSeries Based Linux	Big
Linux 64-bit for AMD	Little
Apple Mac OS	Big
Microsoft Windows 64-bit for AMD	Little
Solaris Operating System (x86 and X86-64)	Little
IBM Power Based Linux	Big

- Use the following command to check the Endianness for your platform:

```
select platform_id, platform_name from v$database;
```

Objects

Some objects cannot be migrated with certain techniques, as shown in the following examples:

- Streams cannot handle secure files Character Large Object (CLOB), National Character Large Object (NCLOB), Binary Large Object (BLOB), and other types. For more information, see “Migration techniques” on page 159).
- Export/Import Data Pump utilities cannot be used for XML types.

8.2.3 Application compatibility

Although Oracle database is supported on System z, you need to check the supported configuration when this database is embedded into an application.

For example, to date, Oracle database as part of Oracle eBusiness Suite is supported on Linux on System z, whereas Oracle database as part of Oracle Financial Services (formerly I-FLEX) is not supported.

> **Tip:** Always check the supported configuration on Oracle Technology Network to avoid the support issues.

8.2.4 In-house administration scripts

You might have in your company a set of scripts to administer your Oracle databases and automate backup and recovery operations, for example. These scripts might have to be customized and should be tested on your new Linux on System z environment.

8.2.5 Network

If the migration technique uses the network (for example, replication techniques), you need to ensure that the network is efficient in terms of bandwidth and latency; otherwise, this potential bottleneck dramatically increases the migration operation duration.

The chosen technique must also take into account the location of the servers, target, and source. Constraints can include that they are geographically dispersed or they cannot communicate with each other.

8.2.6 Disk space requirement

Some migration techniques need the disk space for staging or dump files. You should consider this if you have storage constraints, as shown in the following examples:

- If the migration is done by using Export/Import, you need some space on the source system to store the dump files.
- If the migration is done by using Recovery Manager (RMAN), you need some space to store the redo logs that are created after the copy starts.

8.2.7 Skills

A migration can be considered a risky operation. Depending on the products and techniques that are already used in your environment, you might prefer one technique over another.

> **Tip:** Whenever possible, perform the migration with known products to mitigate the risks.

8.3 Migration techniques

In this section, we describe the available techniques to migrate an Oracle database on Linux on System z.

The following migration techniques are available:

- Export/Import with Oracle Data Pump utilities. For more information and to see an example, see “Example of migration by using Export/Import Data Pump” on page 171.
- Transportable Tablespaces
- Create Table As Select (CTAS)
- Streams Replication
- Oracle GoldenGate
- Transportable Database
- Data Guard Heterogeneous Primary and Physical Standbys
- IBM InfoSphere® Data Replication Server
- IBM XenoBridge

This list is not exhaustive and we describe the most used techniques in following sections.

8.3.1 Export/Import with Oracle Data Pump

Export/Import is the classic method that is used to migrate a database. The Export/Import Data Pump utility is more efficient than the standard Export/Import utility, but can be used starting with Oracle 10g version only. When possible, we recommend using Data Pump. This section describes only the Oracle Data Pump utility.

Export and Import utilities transfer data objects between two databases, independently from hardware and software configurations. Objects can be tables, indexes, comments, grants, and so on.

With Export, objects are extracted (tables first, then other objects, if any) and the extracted data is written into an Export dump file (Oracle binary-format). The Import utility takes the tables data and definitions from the dump file.

Main migration steps

This technique includes the following steps:

1. Export the database with Export utility from the source (the dump file can be on a disk or on a tape).
2. Transfer the dump file to the target by using FTP, SFTP, RCP, or physically if there is no communication between the source and target servers (for example, dump files on tape).
3. Create the target database.
4. Import the data with Import utility in the new database.
5. Import the metadata to complete the full database structure.
6. Check the consistency.

Advantages

This technique features the following advantages:

- Can be used across any platform
- No conversion is needed
- Parallelism helps speed up the operation
- Fine-grained object selection for inclusion or exclusion
- Ability to restart without loss of data
- Database can be migrated and upgraded in one operation
- Space estimation from a storage point of view

Limitations

This technique includes the following limitations:

- Cannot be used with database versions below Oracle 10g. For older versions, use a standard Export/Import. It is slower and has the following restrictions:
  - BINARY_DOUBLE and BINARY_FLOAT data types cannot be exported with EXP utility
  - Java classes, resources, and procedures that are created with Enterprise JavaBeans are not processed
  - Data is not stored in the compressed format when it is imported
- Dump files that are generated by the Data Pump Export utility are not compatible with dump files that are generated by the original Export utility
- Downtime can be significant for large databases

8.3.2 Transportable Tablespaces

The Transportable Tablespace migration works within the framework of Data Pump and RMAN.

RMAN is a backup and recovery manager that is provided by Oracle and does not require any separate installation. RMAN stores metadata in the control files of the target database and, optionally, in a recovery catalog schema in an Oracle database. RMAN provides block-level corruption detection during backup and restore.

Main migration steps

This technique includes the following steps:

1. Convert tablespaces into read-only.
2. Use Data Pump to move the metadata of the objects.
3. Convert the tablespaces to the correct Endianness (if necessary) with RMAN Convert.
4. Create the database on the target system.
5. Transfer the data files to the target server.
6. Import the tablespaces.
7. Import the metadata.
8. Check the consistency.

Advantage

The main advantage in using this technique is that it can be used across different Endianness. If Endianness are the same, we can use the Transportable Database feature, as described in “Other techniques” on page 164.

Limitations

This technique includes the following limitations:

- Requires a larger time investment to test the migration and to develop methods of validating the database and application. Consider whether the additional testing time, complexity, and risk that is involved are worth the potential to reduce migration downtime.
- Requires a higher level of skills for the database administrator and application administrator that is compared to the use of Data Pump full database Export and Import.
- Does not transport objects in the SYSTEM tablespace or objects that are owned by special Oracle users, such as, SYS or SYSTEM. Applications that store some objects in the SYSTEM tablespace or create objects as SYS or SYSTEM require more steps and increase the complexity of the platform migration.
- Self-contained Oracle TableSpaces can be moved between platforms only.
- If the destination database already contains a tablespace with the same name, you must rename or drop it.
- Triggers, packages and procedures must be re-created on the target database.
- Only user tablespaces can be transported. System and SYSAUX objects must be created at the target.
- Tablespaces must be self-contained. (Materialized views or contained objects, such as, partitioned tables, are not transportable unless all of the underlying or contained objects are in the tablespace set.)
- The source and target databases must have the same character set.
- All system privileges are not imported into the upgraded database.
- Resetting sequences and recompiling invalid objects might be needed.
- The Transportable Tablespaces migration approach does not allow for re-architecture of the database (logical and physical layout) as part of the migration.
- Fragmented data is still exists.

8.3.3 Create Table As Select

By using the Create Table As Select technique, you can copy data from a source database to a target database via network.

Main migration steps

This technique includes the following steps:

1. Create a database link.
2. Copy tables between databases by using the **`create table as select`** command.
3. Extract the indexes and constraints by using the **`dbms_metadata`** command.
4. Add them on the target platform.

Advantages

This technique features the following advantages:

- There is no need for extra space for dump files because you copy directly from the source to the target.
- This technique can be used for one or several large tables.

Limitations

This technique includes the following limitations:

- Because this technique uses the network, the network traffic can be significant and slow down other operations, depending on the size of the tables.
- This can be used to migrate one or several tables, but we cannot envisage to use it for an entire database.

This technique can be used with other techniques (Export/Import or Transportable Tablespaces, for example).

8.3.4 Oracle Streams

Oracle Streams uses log data that is captured with LogMiner-based technology on the source system as its capture mechanism, from which logical change records (LCR) are generated. A stream allows transactions to be propagated to one or several databases. The Streams setup uses processes and databases objects to share data and messages. Oracle Streams can specify rules at multiple levels of granularity: database, schema, and table. Oracle Streams capture changes from the redologs in a source database, the changes are staged, and then are propagated into the target database.

This product is used to propagate information among distributed databases, but the mechanism can be used for migration as well with other techniques (Export/Import for example).

Main migration steps

This technique includes the following steps:

1. Create an empty database.
2. Use Oracle Streams to configure a replication environment in which this new database is the target database and the original database is the source database. After populating this new database with, for example, Export/Import, Oracle Streams ensures that Data Manipulation Language (DML) and Data Definition Language (DDL) changes occurring at the source database are ultimately applied at the target database and that the source database is available to users during the operation.
3. When the instantiation is over, use Oracle Streams to apply changes that were made at the source database to the target database.
4. After the changes are successfully applied to the target, this database becomes the database that is available to users.

Advantages

This technique includes the following advantages:

- Minimal downtime (reconnecting the users only).
- Works across platforms without conversions.
- Works across version since Oracle 9iR2.
- Failback is possible because the source is untouched.

Limitations

This technique includes the following limitations:

- Some setup activity is required.
- Some data types are not supported for capture processes, so an Export/Import of the following object types also is required:
  - SecureFile CLOB, NCLOB, and BLOB
  - BFILE
  - Rowid
  - User-defined types (including object types, REFs, arrays, and nested tables)
  - XMLType stored object relationally or as binary XML
  - The following Oracle supplied types:
    - Any types
    - URI
    - Spatial
    - Media

8.3.5 Oracle GoldenGate

By using Oracle GoldenGate, you can move data between like-to-like and heterogeneous systems, including different versions of Oracle Database, different hardware platforms, and between Oracle and non Oracle databases. The software performs real-time, log-based change data capture (CDC) and can move large volumes of transactional data between heterogeneous databases with low latency and minimal footprint.

Main migration steps

This technique includes the following steps:

1. Create an empty database on the target system
2. Install and configure Oracle GoldenGate on the source system and on the target system to capture and deliver changes.
3. During the initialization of the database copy, Oracle GoldenGate persists all new user transactions within its queuing system, which are called Trail Files.
4. After the copy is complete, any new or in-flight transactions that occurred continue to be applied by using Oracle GoldenGate to keep the target database in sync.
5. The current database continues to be active and support users. The new database has all of the current data and is ready for user cutover.
6. After switching over, the active (target) database and the old database can be kept in sync to offer failback option.

Advantages

This technique includes the following advantages:

- Near zero downtime
- Works across platforms without conversion
- Failback is possible

Limitations

This technique includes the following limitations:

- Associated extra license costs
- Memory and CPU overhead (3% - 5% CPU impact of Oracle GoldenGate Replication on the source system, depending on the number of redo logs that are generated)
- The following data types are not supported:
  - ORDDICOM
  - ANYDATA
  - ANYDATASET
  - ANYTYPE
  - BFILE
  - MLSLABEL
  - TIMEZONE_ABBR
  - TIMEZONE_REGION
  - URITYPE
  - UROWID

8.3.6 Other techniques

These techniques are listed but not described in detail because they are not used as often to migrate an Oracle Database onto System z.

Transportable database

With transportable database, we can transport an entire database (user data and the Oracle directory) to a platform with the same Endian format.

This technique uses Recovery Manager (RMAN) and is similar to transportable tablespaces, but the main limitation is that the Endian must be the same for the source and the target.

For more information, see Chapter 10 of *Experiences with Oracle Solutions on Linux for IBM System z,* SG24-7634.

Data Guard Heterogeneous Primary and Physical Standbys

Data Guard depends on the Log Writer process or the Archive process to capture and send redo data or logs to the standby site. This technique can be efficient for other platforms, but there are strong limitations for the System z platform in terms of cross platform compatibility. For System z, the supported migrations are with IBM System z and IBM Linux platforms only.

For more information, see the My Oracle Support note *Dataguard support for Heterogeneous Primary and Physical Standbys in Same Dataguard Configuration* [ID 413484.1].

IBM InfoSphere Data Replication Server

IBM InfoSphere Data Replication Server integrates information across heterogeneous data stores in real time. Among several other capabilities, this product enables zero download database migrations and application upgrades. This product can also replicate data from traditional z/OS data sources, such as, IMS™ and VSAM to DB2®, Oracle, Sybase, and Microsoft SQL Server by combining with InfoSphere Classic Replication Server for z/OS.

For more information, see this website:

http://www-01.ibm.com/software/data/infosphere/data-replication/

Migration with IBM XenoBridge

IBM XenoBridge is a web-based migration tool that uses advanced parallel processing techniques to migrate databases in a minimum of downtime. Tables are moved in parallel (30, 40, or 50 together), XenoBridge can use all of the resources that are available in high-performance servers and SAN disk systems. Combined with an intelligent migration engine and multiple streams, the throughput that is obtained by IBM XenoBridge is up to 30 - 40 times that of export/import.

For more information, contact your IBM representative or the Migration Factory, which is available at this website:

http://www.ibm.com/systems/migratetoibm/factory/

8.3.7 Considerations when migrating from File System to ASM or vice versa

The following organization types are available for your database files:

- File System
- Automatic Storage Management (ASM)

ASM is built into the Oracle kernel and provides the DBA with a way to manage many disk drives for single and clustered instances of Oracle. ASM is a file system/volume manager for all Oracle physical database files (such as, data files, online redo logs, control files, archived redo logs, RMAN backup sets, and SPFILEs). All of the database files (and directories) to be used for Oracle are contained in a disk group.

If you decide to change the way your Oracle database files are organized, you can use RMAN Backup/Restore capabilities.

For more information about migrating databases to and from ASM by using recovery manager, see this website:

http://docs.oracle.com/cd/B14117_01/server.101/b10734/rcmasm.htm

8.4 Best practices

To be successful in your migration operation, we strongly recommend that you review the best practices that are presented here. Most of these practices are based on common sense and some are based on experiences.

The most important best practice is to test.

8.4.1 Sizing considerations

Before any migration begins, we must evaluate the resources that are needed for the Linux for System z guest. If you undersize your target system, you can experience strong performance degradation. For sizing resources, we strongly recommend that you contact your IBM representative and use the IBM Sizing Questionnaires for Oracle Database that is available at the following website to get the best estimation:

http://www.ibm.com/support/techdocs/atsmastr.nsf/PubAllNum/PRS1887

CPU

To evaluate the CPU that is needed (quantity of IFLs), you can ask IBM Techline for a sizing, or to use the IBM SURF and SCON tools if you have access to them.

For this evaluation you need the following information:

- Details for the type of server for your target source (server and type model, number of CPUs and cores, type and speed of cores, and so on)
- The average and peak CPU usage (from NMON or SAR data for Linux, perform for Windows or equivalent product)
- The type of workload (for example: production database)

You can ask your IBM representative to get this evaluation.

Memory

On Linux on System z, it is not recommended to oversize the memory, especially in a virtualized environment.

You need to make sure that the source database memory is optimized. For more information, see the SGA target advisory and PGA target advisory sections of the Oracle AWR reports.

Assuming that the source system memory is optimized, the suggestion is to use the same quantity of memory on the target database as on the source database.

To get a representative Oracle AWR, report you must take your snapshots during the peak period of your workload. You need to determine the peak period within the most loaded day of the week (or month at certain periods). You can run an ADDM report from your Oracle Enterprise Manager Database Control for this purpose.

You can find the memory that is used by Oracle in AWR report in the Memory statistics section, as shown in Figure 8-1.

Memory Statistics

	Begin	End
Host Mem (MB):	96,657.0	96,657.0
SGA use (MB):	19,968.0	19,968.0
PGA use (MB):	1,128.6	1,310.3
% Host Mem used for SGA+PGA:	21.83	22.01

Figure 8-1 AWR Memory statistics section

You can find the quantity of memory that is allocated to the Oracle database in AWR report in the `init.ora` section, as shown in Figure 8-2.

memory_max_target	32212254720
memory_target	32212254720
open_cursors	1000
optimizer_capture_sql_plan_baselines	FALSE
optimizer_index_caching	90
optimizer_index_cost_adj	10
optimizer_mode	ALL_ROWS
parallel_max_servers	4
pga_aggregate_target	2621440000

Figure 8-2 AWR Init.ora section

To make sure SGA and PGA are optimized, you can check the AWR reports advisory section.

For SGA, you find the information in the AWR SGA Target Advisory section, as shown in Figure 8-3.

SGA Target Advisory

SGA Target Size (M)	SGA Size Factor	Est DB Time (s)	Est Physical Reads
7,488	0.38	4,392,392	796,804,239
9,984	0.50	3,929,044	686,043,658
12,480	0.63	3,686,946	628,157,348
14,976	0.75	3,541,351	593,348,453
17,472	0.88	3,437,787	568,783,948
19,968	1.00	3,362,468	550,773,650
22,464	1.13	3,277,398	530,229,793
24,960	1.25	3,228,978	518,663,546
27,456	1.38	3,187,956	508,804,698
29,952	1.50	3,174,170	508,804,698
32,448	1.63	3,168,455	508,804,698
34,944	1.75	3,165,428	508,804,698
37,440	1.88	3,164,420	508,804,698
39,936	2.00	3,164,083	508,804,698

Figure 8-3 AWR SGA Target Advisory section

For information about PGA, see the AWR PGA Target Advisory section, as shown in Figure 8-4.

PGA Memory Advisory

- When using Auto Memory Mgmt, minimally choose a pga_aggregate_target value where Estd PGA Overalloc Count is 0

PGA Target Est (MB)	Size Factr	W/A MB Processed	Estd Extra W/A MB Read/ Written to Disk	Estd PGA Cache Hit %
1,344	0.13	43,134,805.68	8,929.04	100.00
2,688	0.25	43,134,805.68	5,731.56	100.00
5,376	0.50	43,134,805.68	4,320.32	100.00
8,064	0.75	43,134,805.68	4,320.32	100.00
10,752	1.00	43,134,805.68	4,320.32	100.00
12,902	1.20	43,134,805.68	4,140.49	100.00
15,053	1.40	43,134,805.68	4,140.49	100.00
17,203	1.60	43,134,805.68	4,140.49	100.00
19,354	1.80	43,134,805.68	4,140.49	100.00
21,504	2.00	43,134,805.68	4,140.49	100.00
32,256	3.00	43,134,805.68	4,140.49	100.00
43,008	4.00	43,134,805.68	4,140.49	100.00
64,512	6.00	43,134,805.68	4,140.49	100.00
86,016	8.00	43,134,805.68	4,140.49	100.00

Figure 8-4 AWR PGA Memory Advisory section

For the dedicated server processes, you can calculate the memory that is needed as shown in the following example:

```
Memory needed for dedicated server processes = Max(logons concurrent) X memory
used per thread
```

(On average, dedicated connections use 4.5 MB per connection, which is application dependant.)

You can find the concurrent log ons in AWR report, in the Instance Activity Stats - Absolute Values section, as shown in Figure 8-5.

Instance Activity Stats - Absolute Values

- Statistics with absolute values (should not be diffed)

Statistic	Begin Value	End Value
session pga memory max	111,875,419,168	112,985,112,304
session cursor cache count	9,585,942	9,604,038
session uga memory	182,252,173,454,424	182,467,047,728,936
opened cursors current	1,572	1,663
workarea memory allocated	71,349	106,011
logons current	372	400
session uga memory max	776,193,213,352	777,561,241,408
session pga memory	77,531,681,504	77,820,229,232

Figure 8-5 AWR Instance Activity Statistics Absolute Value section

I/O information

You can find the I/O information in AWR reports in the load profile section, as shown in Figure 8-6 on page 169. Physical reads and Physical writes values help you to size the I/O for this workload on System z.

Load Profile

	Per Second	Per Transaction	Per Exec	Per Call
DB Time(s):	2.3	0.0	0.00	0.00
DB CPU(s):	1.4	0.0	0.00	0.00
Redo size:	577,644.1	9,760.7		
Logical reads:	72,874.3	1,231.4		
Block changes:	1,712.5	28.9		
Physical reads:	523.1	8.8		
Physical writes:	95.1	1.6		
User calls:	5,399.5	91.2		
Parses:	2,636.1	44.5		
Hard parses:	10.2	0.2		
W/A MB processed:	11.2	0.2		
Logons:	0.2	0.0		
Executes:	9,043.6	152.8		
Rollbacks:	40.8	0.7		
Transactions:	59.2			

Figure 8-6 AWR Load Profile section

8.4.2 Performance measurement before and after migration

To compare the performance before and after migration, we advise you to take real performance measurements. You want to avoid the possible user complaint that the system was running faster before the migration to this new system without having real figures to compare against. You can get the following performance measurements:

- Operating system level:
  - SAR
  - NMON
  - VMSTAT
  - IOSTAT
- Oracle database level: AWR reports
- Application level:
  - Duration of some batches
  - Response time for some complex user transactions

These performance measurements provide the baseline before any migration/operation.

After the database is migrated to the target system, we recommend that you repeat the performance measurements on the target system to compare the results before and after. If some performance issues appear, the new system needs some tuning. For more information, see Chapter 7, “Tuning z/VM, Linux, and Oracle to run on IBM System z” on page 137.

8.4.3 Real Application Testing feature

Real Application Testing is a feature that is provided by Oracle starting with version 11gR2 that allows to capture and to replay the production workload. Because of the SQL Performance Analyzer feature, Real Application Testing assesses the effect of changes on the tested workload. This feature can minimize the risks of migration and is proactive in terms of performance issues detection.

8.4.4 Preparing the data to be migrated

Before any migration, we recommend that you compress the database as much as you can (reorg, data compression), which reduces the data to migrate and the possible downtime.

8.4.5 Determine the invalid objects in the database

Before migration, you must determine the invalid objects so that you can compare this figure after the migration. This list of invalid objects facilitates any diagnostic tasks.

Use the following command to determine the invalid objects:

```
SELECT object_name, owner, object_type
FROM dba_objects
WHERE status='INVALID'
ORDER BY object_type;
```

8.4.6 Rebuilding indexes

To reduce the migration duration (especially if you choose Export/Import utilities), we recommend that you limit the quantity of data to migrate. For example, you can import in two operations: import of the data and import of the metadata.

This significantly reduces the migration time, mainly because the indexes and the data are not imported at the same time.

8.4.7 Logging during migration

If you plan to use Export/Import for your migration, we suggest you use one of the following methods to disable the logging for archiving:

- Disable the archive logs during the import, if possible (for this, the database must be stopped to be placed in “mount” state).
- Disable the logging directly in to the tablespace by using the **`alter tablespace...nologging`** command.
- Set the parameters _disable_logging bolean TRUE in the `init.ora`.

8.4.8 Redo Logs

We also suggest the following tasks to simplify the Redo logs management:

- Minimize the number of Redo log members per thread (one member is sufficient).
- Increase the size of the Redo logs if possible (to avoid useless switches).

8.4.9 Using Linux on System z infrastructure

To get the best level of performance, you need the same degree of parallelism for Export and for Import (number of processors parameter).

In most of the cases, if you run a consolidation project, you have more processors on your source system than on your Linux guest on System z. You can then use the flexibility of your Linux on System z infrastructure by allocating more CPU and memory during the import phase.

When the import is over, you can adjust the configuration according to your Linux guests needs.

8.4.10 Considerations when you are migrating from Oracle on z/OS to Oracle on Linux on System z

In this section, we describe the considerations about the character set and Export/Import with Data Pump utilities.

Character set

z/OS does not have the same character set as Linux; z/OS is EBCDIC whereas Linux is ASCII. Unicode solves code page mapping issues. For more information, see *My Oracle Support note: Choosing a database character set means choosing Unicode*, 333489.1.

Export/Import with Data Pump utilities

Data Pump utilities are available, but they use UNIX System Services, so we recommend using the Export utility on z/OS.

8.5 Example of migration by using Export/Import Data Pump

In this section, we describe each step of a migration that is done by using Export/Import Data Pump, which is the most commonly used technique for migrating a database.

8.5.1 Infrastructure

The source infrastructure is IBM AIX 6.1/Power 7 based. The source DB version is Oracle 11gR2.

The target infrastructure is SUSE Linux Enterprise Server 11 SP1 on zEnterprise. The target DB version is Oracle 11g R2.

8.5.2 Tasks list

Migration can be a complex project with many tasks and several stakeholders. To make sure that your migration is a success, we recommend that you list the tasks to perform with the associated planning and defined role name for each task.

Our example is simple and the main steps are shown in Table 8-2 on page 172.

Table 8-2 Migration list of steps example

Task	Location	Role name
Snapshot of DB status	Source	DBA
Export source DB	Source	DBA
Transfer dumpfile to target	Source	DBA
Install Oracle binaries	Source	DBA
Import source DB	Target	DBA
Open DB	Target	DBA
Check snapshot	Target	DBA

8.5.3 Snapshot of the database status

The idea of the snapshot is to provide an overview of what is migrated and verify the success of the operation. Complete the following steps to obtain a snapshot of the database status:

1. List the users in the database, as shown in the following example:

```
SQL> select username, user_id from dba_users order by user_id;
   USERNAME                          USER_ID
   ------------------------------ ----------
   SYS                                     0
   SYSTEM                                  5
   OUTLN                                   9
   DIP                                    14
   ORACLE_OCM                             21
   DBSNMP                                 30
   APPQOSSYS                              31
   WMSYS                                  32
   EXFSYS                                 42
   CTXSYS                                 43
   XDB                                    45

   USERNAME                          USER_ID
   ------------------------------ ----------
   ANONYMOUS                              46
   ORDSYS                                 53
   ORDDATA                                54
   ORDPLUGINS                             55
   SI_INFORMTN_SCHEMA                     56
   MDSYS                                  57
   OLAPSYS                                60
   MDDATA                                 64
   SPATIAL_WFS_ADMIN_USR                  66
   SPATIAL_CSW_ADMIN_USR                  69
   SYSMAN                                 71

   USERNAME                          USER_ID
   ------------------------------ ----------
   MGMT_VIEW                              73
   FLOWS_FILES                            74
   APEX_PUBLIC_USER                       75
```

```
APEX_030200                            77
OWBSYS                                 78
OWBSYS_AUDIT                           82
SCOTT                                  83
SOE                                    85
SH                                     86
XS$NULL                        2147483638
32 rows selected.
```

2. Identify the users that you are migrating. In our case, we are interested in SOE and SH.
3. Check the number of objects of each type before and after the migration, as shown in the following example:

```
select owner, object_type, count(*) from dba_objects
where owner in ('SOE','SH')
group by owner ,object_type order by 1,2

OWNER                          OBJECT_TYPE           COUNT(*)
------------------------------ ------------------- ----------
SH                             INDEX                       10
SH                             TABLE                        8
SOE                            INDEX                       23
SOE                            PACKAGE                      1
SOE                            PACKAGE BODY                 1
SOE                            SEQUENCE                     2
SOE                            TABLE                        9
SOE                            VIEW                         2

8 rows selected.
```

4. Check the number of invalid objects, as shown in the following example:

```
select owner ||' - '|| object_name ||' - '|| object_type from dba_objects where
status = 'INVALID';
no rows selected
```

This information is important, even for those objects that are not included in the previous schemas.

5. Check the grants for each user, as shown in the following example:

```
select  'grant '||granted_role||' to '||grantee||';'
from dba_role_privs
where grantee ='SOE'
UNION
select
'grant '||privilege||' on '||owner||'.'||table_name
||' to '||grantee||';'
from dba_tab_privs
where grantee ='SOE'
UNION
select
'grant '||privilege||' ('||column_name
||') on '||owner||'.'||table_name
||' to '||grantee||';'
from dba_col_privs
where grantee ='SOE';

'GRANT'||PRIVILEGE||'TO'||GRANTEE||';'
```

```
--------------------------------------------------------------------------------
-
grant ALTER SESSION to SOE;
grant CONNECT to SOE;
grant CREATE MATERIALIZED VIEW to SOE;
grant CREATE VIEW to SOE;
grant EXECUTE on SYS.DBMS_LOCK to SOE;
grant QUERY REWRITE to SOE;
grant RESOURCE to SOE;
grant UNLIMITED TABLESPACE to SOE;

8 rows selected.
```

The results from the script give you the commands that are required to re-create any missing grants. However, they must be performed from the correct user.

For example, the **`grant execute on SYS.DBMS_LOCK`** command must be performed from the user `sys`.

8.5.4 Export source database

Complete the following steps to export the database:

1. Before `Expdp` is run, we must position several parameters, including Parallelism degree.

   We check the number or CPUs on both the source and target machines by using the following Linux command:

   ```
   [oracle@oracle11G ~]$ cat /proc/cpuinfo | grep processors
   # processors    : 3
   ```

   In our case, it is three on the target and eight on the source, so we can perform an export with PARALLEL=6 (two times the smaller number of CPUs): dump directory.

   We need to create the dump directory at OS level for the output files and tell Oracle where to put them, as shown in the following example:

   ```
   sqlplus / as sysdba
   SQL > create directory dmpdir as '/u01/app/oracle/export';
   ```

   The name of the directory is passed to `Expdp` as the DIRECTORY parameter.

   FILESIZE is the maximum size of the export files. In this case, we set it to 32 GB, but it must fit with the maximum file sizes on the source and target system.

2. Start the Export utility, as shown in the following example:

   ```
   expdp userid=\"/ as sysdba\" DUMPFILE=soe%U.dmp DIRECTORY=dmpdir FULL=Y
   PARALLEL=6 FILESIZE=32G
   ;;;
   Export: Release 11.2.0.3.0 - Production on Wed Sep 19 17:52:08 2012

   Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.
   ;;;
   Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit
   Production
   With the Partitioning, OLAP, Data Mining and Real Application Testing options
   Starting "SYS"."SYS_EXPORT_FULL_01":  userid="/******** AS SYSDBA"
   DUMPFILE=soe%U.dmp DIRECTORY=dmpdir FULL=Y PARALLEL=6 FILESIZE=32G
   Estimate in progress using BLOCKS method...
   Processing object type DATABASE_EXPORT/SCHEMA/TABLE/TABLE_DATA
   ```

```
Total estimation using BLOCKS method: 2.186 GB
. . exported "SOE"."ORDERS"                             225.7 MB 4513789 rows
Processing object type DATABASE_EXPORT/TABLESPACE
Processing object type DATABASE_EXPORT/PROFILE
Processing object type DATABASE_EXPORT/SYS_USER/USER
Processing object type DATABASE_EXPORT/SCHEMA/USER
Processing object type DATABASE_EXPORT/ROLE
Processing object type DATABASE_EXPORT/GRANT/SYSTEM_GRANT/PROC_SYSTEM_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/GRANT/SYSTEM_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/ROLE_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/DEFAULT_ROLE
Processing object type DATABASE_EXPORT/SCHEMA/TABLESPACE_QUOTA
Processing object type DATABASE_EXPORT/RESOURCE_COST
Processing object type DATABASE_EXPORT/TRUSTED_DB_LINK
. . exported "SH"."CUSTOMERS"                           402.6 MB 2000000 rows
Processing object type DATABASE_EXPORT/SCHEMA/SEQUENCE/SEQUENCE
. . exported "SOE"."CUSTOMERS"                          250.9 MB 4007543 rows
******************************************************************************
Dump file set for SYS.SYS_EXPORT_FULL_01 is:
  /u01/app/oracle/export/soe01.dmp
  /u01/app/oracle/export/soe02.dmp
  /u01/app/oracle/export/soe03.dmp
  /u01/app/oracle/export/soe04.dmp
  /u01/app/oracle/export/soe05.dmp
  /u01/app/oracle/export/soe06.dmp
Job "SYS"."SYS_EXPORT_FULL_01" successfully completed at 17:56:55
```

The export process creates the following output files:

- export.log
- soe01.dmp
- soe02.dmp
- soe03.dmp
- soe04.dmp
- soe05.dmp
- soe06.dmp

It is a recommended best practice that you compress the files to reduce the transfer time.

3. Check the target availability, as shown in the following example:

```
$ ping 10.3.58.126
PING 10.3.58.126 (10.3.58.126): 56 data bytes
64 bytes from 10.3.58.126: icmp_seq=0 ttl=63 time=0 ms
...
```

4. Transfer the files, as shown in the following example:

```
$ sftp 10.3.58.126
The authenticity of host '10.3.58.126 (10.3.58.126)' can't be established.
RSA key fingerprint is 10:00:25:b6:de:9a:50:d8:15:c9:b9:b6:d5:cd:f0:a5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.3.58.126' (RSA) to the list of known hosts.
oracle@10.3.58.126's password:
Connected to 10.3.58.126.
sftp> cd /opt/oracle/export
sftp> mput *.dmp
Uploading soe01.dmp to /opt/oracle/export/soe01.dmp
soe01.dmp                                     100%  312MB  26.0MB/s   00:12
```

```
Uploading soe02.dmp to /opt/oracle/export/soe02.dmp
soe02.dmp                                      100%  541MB  25.8MB/s   00:21
Uploading soe03.dmp to /opt/oracle/export/soe03.dmp
soe03.dmp                                      100%  339MB  26.1MB/s   00:13
Uploading soe04.dmp to /opt/oracle/export/soe04.dmp
soe04.dmp                                      100%  243MB  27.0MB/s   00:09
Uploading soe05.dmp to /opt/oracle/export/soe05.dmp
soe05.dmp                                      100%  303MB  25.2MB/s   00:12
Uploading soe06.dmp to /opt/oracle/export/soe06.dmp
soe06.dmp                                      100%   25MB  24.7MB/s   00:01
sftp> bye
```

8.5.5 Creating a target database

For more information about and to see an example of creating a database, see Appendix B, "Installing Oracle and creating a database 11.2.0.3 on Red Hat Enterprise Linux 6" on page 335.

8.5.6 Import on the target environment

We use the same level of parallelism as the export. Complete the following steps:

1. Create the directory in the database, as shown in the following example:

   ```
   SQL> create directory dmpdir as '/opt/oracle/export';
   Directory created.
   ```

2. If the structure of the data file file systems is not the same, create the new tablespaces, as shown in the following example:

   ```
   CREATE TABLESPACE "SOE" DATAFILE
   '/opt/oracle/oradata2/swingbench/soe/SOE/soe.dbf' SIZE 3436183552 AUTOEXTEND ON
   NEXT 67108864 MAXSIZE 32767M LOGGING ONLINE PERMANENT BLOCKSIZE 8192 EXTENT
   MANAGEMENT LOCAL UNIFORM SIZE 1048576 ;

   CREATE TABLESPACE "SH" DATAFILE
   '/opt/oracle/oradata2/swingbench/soe/SOE/sh.dbf' SIZE 1825570816 AUTOEXTEND ON
   NEXT 67108864 MAXSIZE 32767M LOGGING ONLINE PERMANENT BLOCKSIZE 8192 EXTENT
   MANAGEMENT LOCAL UNIFORM SIZE 1048576 ;
   ```

3. Start the `impdp` utility, as shown in the following example:

   ```
   impdp userid=\"/ as sysdba\" DUMPFILE=soe%U.dmp DIRECTORY=dmpdir FULL=Y
   PARALLEL=6
   impdp userid=\"/ as sysdba\" DUMPFILE=soe%U.dmp DIRECTORY=dmpdir FULL=Y
   PARALLEL=6 CONTENT=METADATA_ONLY
   Job "SYS"."SYS_IMPORT_FULL_01" completed with 8122 error(s) at 16:42:41
   ```

4. Check the number of objects of each type after the migration, as shown in the following example:

   ```
   select owner, object_type, count(*) from dba_objects
   where owner in ('SOE','SH')
   group by owner ,object_type order by 1,2;
   OWNER                          OBJECT_TYPE           COUNT(*)
   ------------------------------ ------------------- ----------
   SH                             INDEX                       10
   SH                             TABLE                        8
   SOE                            INDEX                       23
   ```

```
SOE                            PACKAGE                      1
SOE                            PACKAGE BODY                 1
SOE                            SEQUENCE                     2
SOE                            TABLE                        9
SOE                            VIEW                         2
8 rows selected.
```

5. Check the grants for each user, as shown in the following example:

```
select  'grant '||privilege||' to '||grantee||';'
from dba_sys_privs
where grantee ='SOE'
UNION
select  'grant '||granted_role||' to '||grantee||';'
from dba_role_privs
where grantee ='SOE'
UNION
select
    'grant '||privilege||' on '||owner||'.'||table_name
||' to '||grantee||';'
from dba_tab_privs
where grantee ='SOE'
UNION
select
     'grant '||privilege||' ('||column_name
    ||') on '||owner||'.'||table_name
    ||' to '||grantee||';'
from dba_col_privs
where grantee ='SOE'
;
'GRANT'||PRIVILEGE||'TO'||GRANTEE||';'
--------------------------------------------------------------------------------
-
grant ALTER SESSION to SOE;
grant CONNECT to SOE;
grant CREATE MATERIALIZED VIEW to SOE;
grant CREATE VIEW to SOE;
grant QUERY REWRITE to SOE;
grant RESOURCE to SOE;
grant UNLIMITED TABLESPACE to SOE;
7 rows selected.
```

You see there is one grant that is missing, as shown in the following example:

```
grant EXECUTE on SYS.DBMS_LOCK to SOE;
Grant succeeded.
```

This was an example of a simple migration. There were no database links, no scheduled jobs, and no external tables.

8.6 Summary

In the Table 8-3, we summarize the features and the limitations of most of the techniques that were described in this chapter. This overview can be a good starting point to evaluate what you can and cannot use in your organization.

Table 8-3 Migration techniques summary

Technology	Complexity	Dump/Staging space requirements	Downtime	Limitations
EXP/IMP Data Pump	Simple	Need space for dump files	Significant: Depends on database size	Only for 10g version and above; otherwise, use standard EXP/IMP
Transportable Tablespaces	Complex	Need space for metadata and converted files	Significant: Depends on database size	Only for 10g version and above
Create Table As Select	Complex for a full database	No extra space needed	Significant: Depends on database size	Table-by-table
Streams	Complex	Need space for repository database	Minimum	Additional repository database needed. Some data types are not supported
Oracle GoldenGate	Complex	Need some space for the archive logs	Minimum	Additional product required. Some data types are not supported
Transportable Database	Simple	No extra space needed	Significant: Depends on database size	Same Endianness required
Oracle Data Guard	Simple	Need space for Archive logs	Minimum	Cross-platform limitations

Tip: Testing is the key part of the migration. You need several iterations of tests to successfully perform your migration.

9

High Availability and Disaster Recovery environment for Oracle

This chapter is an introduction to planning a highly available and disaster recovery (HADR) environment for Oracle Databases that are running on Linux on System z in a virtualized environment. IBM System z hardware is designed for continuous availability and offers a set of reliability, availability, and serviceability features (RAS).

Oracle Database is one of the leading technologies with built-in High Availability options. The combination of IBM System z and Oracle Database provides a system that is comprehensive, reliable, and capable of deploying highly available environments that offer varying levels of data, application, and infrastructure resilience.

Many tiers of an HADR solution are possible. Oracle recommends Maximum Availability Architecture (MAA) as the best practices' blueprint for an HA environment. The right HADR configuration is a balance between recovery time and recovery point requirements and cost.

Based on our experiences in implementing Oracle on Linux on System z, we provide a road map in this chapter to plan an HADR environment for Oracle databases.

A highly available environment is a combination of technology, coordination across multiple teams, change control, skills, enterprise culture, and operational discipline. This chapter is an introduction to the various technology options that are available to users (in-depth information that is necessary to implement the architectures is not included here). We encourage the reader to review Oracle MAA white papers that are available on the Oracle Technology Network website for more in-depth descriptions about implementing the right solutions for complex environments.

For more information about High Availability, see Chapter 2, "Getting started on a proof of concept project for Oracle Database on Linux on System z" on page 13, Chapter 3, "Network connectivity options for Oracle on Linux on IBM System z" on page 29, and Chapter 6, "Using z/VM Live Guest Relocation to relocate a Linux guest" on page 117.

This chapter includes the following topics:

- High Availability
- Oracle technologies for High Availability
- High Availability with z/VM
- Disaster Recovery solutions
- Summary

9.1 High Availability

A highly available system is designed to eliminate or minimize the loss of service because of planned or unplanned outages. High Availability does not equate to continuous availability (that is, a system with nonstop service). High Availability in general describes the accessibility and uptime of critical business application environments to the users, who experience frustration when their application is unavailable, and they do not care about the complexity of the application or why it is not available. Availability is always measured by the perception of an application's user.

High Availability is a key component of business resiliency. Although hardware technology like in IBM System z are highly reliable, the unplanned outages, such as, operator errors, software problems, application performance issues, and other non-hardware related factors still can make the systems unavailable. The *five nines of availability* term specifies 99.999% uptime, but in any user environment, it might not be necessary to achieve that and have all the applications available always. There might be some critical applications that always must be up and running, and here might be other applications that do not need as much availability requirements.

High Availability solutions always involve redundancy, and the basic rule of High Availability is to identify and remove single points of failure in the architecture.

A user environment can have multiple layers, such as, cloud, user, application, firewall, security, facility, storage, and database layers. All of these layers can be in one or multiple data centers and they can be in one or multiple servers that are running under multiple operating systems.

In this section, we describe achieving High Availability for Oracle databases in a Linux on System z environment only. Figure 9-1 on page 182 shows an architecture in which components in an Oracle database environment are running on a Linux on System z environment.

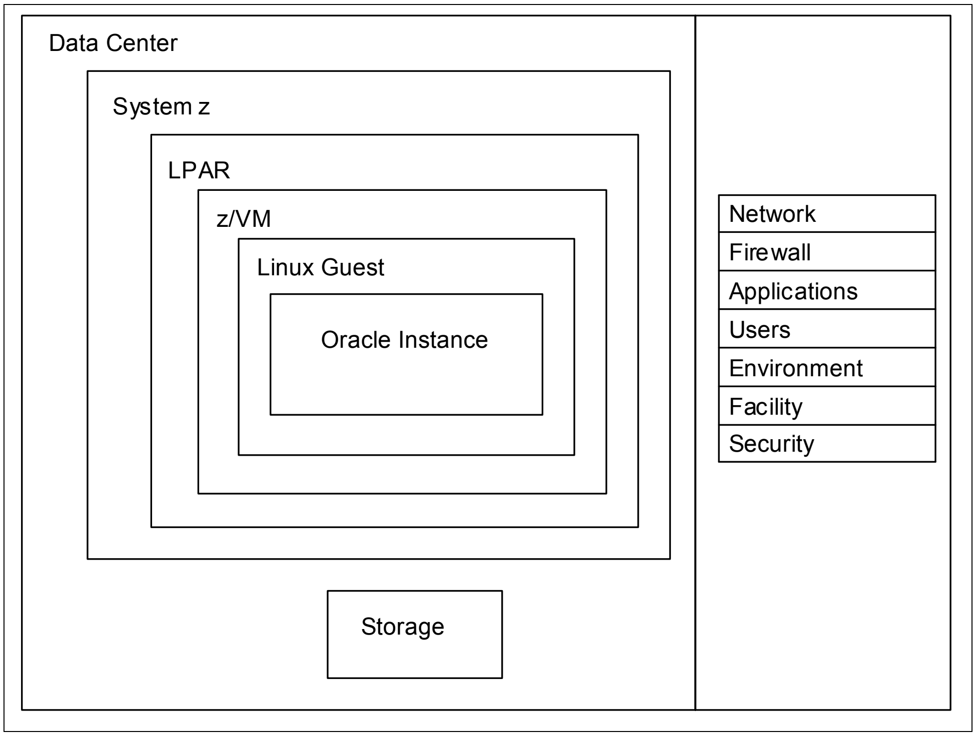


Figure 9-1 Components in Oracle environment on Linux on System z

Data Center

The Data Center is the top layer where all the components that are needed for Oracle databases on Linux on System z are running. This center encompasses all of the servers, storages, network, facility, human resources, software, and other components that are required to run the databases.

Servers

System z server is hardware that provides the computing power (CPU, memory, I/O connections, and power supply).

Logical partitions

System z servers are typically divided into multiple logical partitions (LPARs) to share the System z hardware resources.

z/VM

z/VM is a hypervisor operating system that is running in an LPAR.

Linux guests

z/VM hypervisor that is running in an LPAR can host one or more Linux operating systems in entities that are known as *virtual machines*. We refer to a virtual machine that is running an operating system as *guest*. Also, Linux operating system can run natively in an LPAR.

Oracle instances

In a Linux guest, one or more Oracle instances are running.

Disk Storage

z/VM, Linux, and Oracle need non-volatile storage for their operations, in the same way that Oracle instances have their database files on the storages.

In the environment that is shown in Figure 9-1 on page 182, there can be failures in any one of the components, which can cause unavailability. Planned or unplanned downtime is costly and in the following sections we describe the general causes for planned and unplanned outages in the environment that is shown in Figure 9-1 on page 182.

9.1.1 Planned downtime events

In general, the planned downtime in an Oracle environment can be started by any of the following events:

- Software maintenance and upgrades
- Hardware refreshes
- Data center relocations
- Building maintenance

It might be surprising to note that the largest share of time that a database is rendered unavailable is because of the planned maintenance activities.

In many situations, the planned downtime activities can be coordinated with users in advance. With the understanding of the business requirements and proper planning, the Service Level Agreements (SLA) can be met and the effect to the user community can be minimized.

9.1.2 Unplanned downtime triggers

In any environment, unplanned downtime activities can result in prolonged application failures, and they might result in unsatisfied users and revenue loss. Any of the reasons that are described in this section might trigger downtime in an Oracle environment and can jeopardize the SLA.

Data Center

The Data Center where the systems are deployed might not be available because of any of the following factors:

- Natural disasters
- Sabotages
- Power failures
- Network or firewall failures

Hardware components

The hardware components that are hosting the applications might not be available for any of the following reasons:

- Hardware failures (CPU, memory, power supply, cables, storage, or switches)
- Bottleneck of resources (CPU, memory, storage, or network)

System software components

Any of the following system software components can fail:

- Hypervisor (z/VM)
- Operating systems (Linux)
- Monitoring software

Application components

The applications, load balancers, web servers, and other associated software components might fail because of the following factors:

- Application Logic
- Application overload

Oracle database

Oracle databases might not be available for any of the following reasons:

- Instances failure
- Components (Listeners) are not available
- Data files corruption or deletion
- Logical data corruptions
- Security violations
- Administration issues (file sizes cannot be extended)
- Performance issues
- Scalability issues (additional users)
- Critical patches application requirements

9.1.3 Defining the common requirements for High Availability

Any of the events that were described might result in unavailability. Any HADR solution should ensure the following resiliencies:

- Infrastructure resiliency

  Normally, infrastructure resiliency is obtained by reliable, redundant, and clustered components.

- Data resiliency

  Data is the critical component of any system, and the resiliency of the data availability is a minimum requirement for any HADR solution. Storage mirroring and database technologies drive the resiliency of the data.

- Application resiliency

  In an ideal HADR design, when the system is recovered for operations, the users must be able to continue where they left off after recovery is complete. This application stateless resiliency is also a basic requirement.

In this chapter, we focus more on data resiliency, and, to a lesser extent, infrastructure resiliency. The application resiliency is beyond the scope of this document.

The effort that is needed to implement a highly available system depends on the following measurements:

- Recovery Time Objective (RTO)

  RTO is the maximum period for which a disturbance to an application or a process can be tolerated before the business or financial effect is unacceptable. RTO is inclusive of issue identification, response, and issue resolve time.

- Recovery Point Objective (RPO)

  RPO is the maximum amount of time data loss can be tolerated. For some applications, the requirement can be zero data loss, as in a stock trading application. In other cases, it can be couple of hours worth of data losses.

- SLA

  SLA is the agreed upon downtime for an application with the user. This might span from system availability to functional availability in the application. Typically, the query response time during online or the reports creation time during batch running dictates this requirement.

In business environments, users also divide their applications into multiple tiers. Typically, Tier 1 applications can have the strictest RTO, RPO, and SLA requirements. Tier 2 and Tier 3 have less stringent requirements. The HADR solutions for Tier 1 applications normally are costlier to implement.

> **Note:** The right High Availability configuration is a balance between the recovery requirements and cost.

9.2 Oracle technologies for High Availability

In an Oracle environment, the undisputable requirement is data resiliency. The data should be available always. If a failure occurs, the data loss should be zero or minimal. In this section, we introduce the various technologies that are available from Oracle to build a highly available environment for Oracle databases.

9.2.1 Backup and recovery

The foundation for any robust Oracle highly available environment is having a solid, reliable backup and recovery process. In some situations, as in the following examples, database backups might be the only recovery options:

- For a block corruption, media failure, or other physical data failures where there is no Data Guard in the environment, the only solution is to restore from existing backups.
- Failures at primary and secondary sites of a Data Guard solution.
- Setting up the initial Data Guard environment.

Backups can be performed at the logical and physical level. An effective backup strategy must be based on physical backups that allow the database to be restored with consistency. Logical backups, such as, exporting database objects, are a useful supplement to physical backups but cannot protect the whole database.

The backups can be consistent or inconsistent. During the consistent backup, the database is shut down cleanly and remains closed during the backup. All committed changes in the online redo log files are written to the data files during the shut down process, so the data files are in a transaction-consistent state. The database can be opened immediately after the data files are restored from the consistent backup.

It is not always possible, however, to shut down the database for backup. In that case, by enabling the ARCHIVELOG for the database, the backups can be taken, which are called inconsistent backups. In this case, the online redo log files contain changes that are not yet applied to the data files. The online redo log files must be archived and then backed up with the data files to ensure recoverability. If the archived redo log files and data files are restored, media recovery must be performed before the database can be opened.

For Oracle Databases backup Oracle's Recovery Manager (RMAN) utility is the ideal choice for most users. Many third-party vendors, such as, IBM Tivoli® integrated with RMAN utilities to offer value-added services for backup and recovery.

Using RMAN for backup includes the following advantages:

- RMAN automatically determines what files are to be backed up and what files must be used for media-recovery operations.
- Online database backups are done without placing tablespaces in backup mode.
- Block-level incremental backups and data block integrity checks are done during backup and restore operations.
- Automated tablespace point-in-time recovery and block media recovery is available.

Best practices for backup and recovery

The following best practices are available for backup and recovery tasks:

- Oracle recommends a *Fast Recovery Area* to simplify the management of backup and recovery files. This area is an Oracle managed directory, file system, or Oracle Automatic Storage Management (ASM) disk group that provides a centralized disk location for backup and recovery files. Oracle also creates archived logs and flashback logs in the fast recovery area. RMAN can store its backup sets and image copies in the fast recovery area, and RMAN uses it when restoring files during media recovery. The fast recovery area also acts as a disk cache for tape. It is important that the fast recovery area is sized according to the database's file sizes, transaction rates, and retention policy requirements.
- Establish block changes tracking, incremental backup processes, and the backup frequency, which should be based on the following factors:
  - The RPO, which depends on the data criticality
  - The RTO, which is dictated by data repair time
  - Database transaction rates for data changes
- Establish backup retention policy:
  - Distinct redundant data files, recovery window-based files, or both policies should be established on the basics of data criticality.
  - The legal requirements might also determine the backup files retention policy.
  - The use of RMAN recovery catalog is a good practice and it is preferred to use the catalog schema in a dedicated stand-alone database.
  - RMAN should be configured to automatically back up the control file and the server parameter file (SPFILE).

HADR applicability

Solid backup and recovery process is the foundation and part of any HADR configuration.

9.2.2 Oracle Flashback Technology

Backup and recovery processes protect failures against the physical deletion of database files, media corruption, and dropping of database logical entities. In any database environment, it is impossible to avoid human errors, such as, an authorized user running an incorrect query and deleting rows in the tables or corrupting some data. Though it is possible to recover the data from the backup files, it can take hours to rebuild the database. Oracle provides a Flashback technology to reverse human errors by selectively undoing the effects of a mistake. Flashback supports recovery at the row, transaction, table, and the entire database levels.

Facilities provided by Oracle Flashback Technology

Oracle Database 11g Release 2 includes enhancements to enable Flashback Database while the database is open. Oracle Flashback Technology can accommodate the reversal of the following types of actions by using flashback logs:

- Flashback database:
  - Entire database to a specific point-in-time can be restored.
  - Flashback Database is fast compared to traditional backup and recovery process because it restores blocks that changed only.
  - The Database can be rewound based on System Change Number (SCN), time stamp, or restore points.
- Flashback drop:

  By using the Flashback Drop feature, dropped tables can be recovered. The dropped table, and all of its indexes, constraints, and triggers, from the Recycle Bin are recovered.
- Flashback table:

  The logically corrupted table can be restored to a specific point. The corrupted table can be rewound, undoing any updates that are made to the table between the current time and the specified time.
- Flashback Query

  By using Oracle Flashback Query, users can query any data at some point in the past. This feature can be used to view and logically reconstruct corrupted data that might be deleted or changed inadvertently.

  This facility allows identification and resolution of logical data corruption.
- Flashback Versions Query

  By using Oracle Flashback Versions Query, users can retrieve different versions of a row across a specified time interval instead of a single point. Users also can pinpoint exactly when and how data changed, which enables data repair and application debugging to identify and resolve logical data corruption.
- Flashback Transaction Query

  A flawed transaction might result in logical data corruption across the tables. Flashback Transaction Query shows the changes that are made by a transaction and also produces the SQL statements necessary to flashback or undo the transaction.
- Flashback Transaction

  A flawed transaction can result in logical data corruption across the tables. With Flashback Transaction, a single transaction (and optionally, all of its dependent transactions), can be flashed back.

  Flashback Transaction relies on undo data and archived redo logs to back out the changes

HADR applicability

HADR applicability includes data corruption that is caused by human errors.

9.2.3 Oracle Automated Storage Manager

Oracle Automated Storage Manager (ASM) is an integrated database file system and disk manager. Oracle ASM groups the disks in the storage system into one or more disk groups, and automates the placement of the database files within those disk groups. It reduces the complexity of managing thousands of files in a large environment. Oracle ASM is part of the Oracle Grid Infrastructure (GI), and it is installed when the Oracle Grid is installed. Oracle clusterware and ASM are installed into the same Oracle home.

ASM features

ASM includes the following features:

- An ASM disk group is a collection of disks that us managed as a unit. A disk group can have as many as 10,000 disks and each disk can have a maximum size of 2 TB.
- Each disk group is self-contained and has its own ASM metadata. An ASM instance manages that ASM metadata.
- In Oracle 11.2 three disk groups are specified: one for data, one for flash recovery and archive, and another one for SPFILE, voting, and Oracle Cluster Registry (OCR).
- In large enterprises, the data disks can be groups that are based on the storage tiers. The best practice is to use similar performance level and similar sized disks within a group. The disk size is not an influential factor, and a minimum of four disks is recommended per group.
- ASM looks for disks in the operating system location that is specified by the `ASM_DISKSTRING` initialization parameter.
- For 11gR2, the SCAN listener is run from GI Home and database listener from DB HOME.
- Oracle recommends RMAN to back up and transport database files in ASM

ASM benefits

ASM features the following benefits:

- ASM spreads data evenly across all disks in a disk group. This software-controlled striping evenly distributes the database files to eliminate the hot spots.
- Optionally, ASM supports two-way mirroring in which each file extent receives one mirrored copy. It also supports three-way mirroring in which each file extent receives two mirrored copies. Additionally, ASM mirrors at file level, and the mirrored copy is kept at a disk other than the original copy disk. This configuration improves the availability.
- Dynamic addition of disks and removal facility of ASM improves the storage availability.
- ASM can now store Voting and OCR files for Oracle clusters.
- ASM reduces administrative tasks by enabling files that are stored in Oracle ASM disk groups to be Oracle Managed Files. It reduces the complexity of managing thousands of files in a large environment.

HADR applicability

HADR applicability includes the following factors:

- Data corruption
- Storage failures

9.2.4 Oracle Grid Control Cluster technology

Oracle Grid Infrastructure technology allows clustering of independent servers so that they cooperate as a single system. If a clustered server fails, any managed application can be restarted on the surviving servers. Oracle Grid Infrastructure software integrates Oracle Clusterware and Oracle ASM and provides the infrastructure necessary for a High Availability framework. The managed applications can be like Siebel, GoldenGate, WebSphere, or even Oracle databases.

Oracle Grid Infrastructure features

Oracle Grid Infrastructure includes the following features:

- Oracle Clusterware provides cluster management capabilities, such as node membership, group services, global resource management, and High Availability functions.
- The applications are protected in active/passive environment.
- For High Availability, applications can be placed under the protection of Oracle Clusterware so that they can be restarted in the primary node when the built-in agent detects the application failure.
- By using built-in agents, if the primary node fails, it can restart the application on the other active nodes in the cluster.
- The monitoring frequency, starting, and stopping of the applications and the application dependencies are all configured.

HADR applicability

HADR applicability includes the following features:

- Protection from Computer hardware failures
- Protection from OS (Linux/z/VM) failures
- Protection from Oracle instance failures
- Protection from storage failures (if ASM is used)
- Active/passive configuration, recovery is not instantaneous

9.2.5 Oracle RAC One Node technology

Oracle Real Application Clusters One Node (Oracle RAC One Node) technology is a new option in 11.2. and provides a fail over solution for Oracle databases. Oracle RAC One Node is a single instance of an Oracle RAC database that runs on one node in a cluster. It uses Omotion technology to relocate the instance without any downtime and does not need manual intervention. During the short period when the instance is moved from one node to another, both instances are active. After all of the connections are migrated, the first instance goes down. If the active instance suddenly fails, Oracle RAC One Node detects the failure and restarts the failed database or fails it over to another server.

Oracle RAC One Node features

Oracle RAC One Node includes the following features:

- Running Oracle instance can be migrated from one server to another without disruption of service.
- Better availability than Clusterware active/passive solution.
- Online patching and upgrading of operating system and database software without downtime is possible.

- Databases can be consolidated into a single cluster for efficient administration. If a server fails, they can be quickly relocated.
- Ready to scale and upgrade to multi-node Oracle RAC for scalability.

HADR applicability

HADR applicability includes the following features:

- Protection from Computer hardware failures
- Protection from OS (Linux/z/VM) failures
- Protection from Oracle instance failures
- Protection from storage failures (when ASM is used)
- For planned outages, it is possible to have continuous availability of Oracle instances

9.2.6 Oracle RAC technology

System z provides a reliable architecture and avoids server as a single point of failure. But it is possible that the Oracle instances, operating systems, such as, Linux, hypervisors, such as, z/VM can fail. These components can introduce the single point of failures, unless clustered.

In a typical Oracle environment, an Oracle instance that is running on a Linux guest under z/VM hypervisors accesses a single database. If the instance stops or the Linux guest where Oracle instance is running fails, access to the data is impossible. The Oracle RAC technology allows multiple Oracle instances that are running across multiple nodes to access the same database and provides a single logical instance view. A cluster in this case can be defined as a pool of independent servers that are acting as a single system.

Oracle Clusterware technology allows clustering of independent servers so that they cooperate as a single system. Oracle Grid Infrastructure software integrates Oracle Clusterware and Oracle ASM and provides the infrastructure that is necessary for High Availability framework. With Oracle RAC, all nodes are active and it enables the continuous availability of Oracle instance.

Oracle RAC features

Oracle RAC includes the following features:

- Ability to tolerate and quickly recover from computer and instance failures.
- Rolling upgrades for system and hardware changes.
- Rolling patch upgrades for some interim patches, security patches, CPUs, and Cluster software.
- Scalability by adding more instances (servers).

Oracle Extended RAC features

Oracle Extended RAC is an architecture in which the nodes in the cluster are separated into different data centers. It provides fast recovery from a site failure and allows for all nodes at all sites to actively process transactions as part of single database cluster. This means that it provides the highest level of availability for server and site failures. It includes the following challenges:

- Redundant connections and sufficient bandwidth for public traffic, interconnect, and I/O.
- High interconnect and network latency can throttle database performance and response time.
- 10 km distance between nodes might require Dark Fiber and, therefore, is high cost.

- Unlike Oracle Data Guard, RAC is a single database (no secondary database) and data corruptions, lost writes, or database-wide failures are possible.
- Storage complexity.

HADR applicability

HADR applicability includes the following features:

- Protection from Computer hardware failures.
- Protection from OS (Linux/z/VM) failures.
- Protection from Oracle instance failures.
- Protection from storage failures (when ASM is used).
- Active/active configuration and hence continuous availability.
- Fast Application Notification (FAN) with integrated Oracle client failover.
- Server side callouts to log trouble tickets or page Administrators to alert them of a failure.
- Complex solution.

Note: A full description of RAC architecture is beyond the scope of this document. For more information, see "Related publications" on page 391.

9.2.7 Oracle Application Failover technology

When a planned or unplanned database outage occurs, the applications can encounter errors or hangs. Oracle's High Availability features address these hangs by providing APIs to speed up the error response and, in some case, mask the error to the users. The database and the application tiers should be configured for fast application failover.

At a high level, automating client failover in an Oracle RAC configuration includes the following steps:

1. Relocate the database services to new or surviving instances.
2. Notify the clients that a failure occurred.
3. Redirect the clients to the relocated or a surviving instance.

FAN

FAN emits events when database conditions change, such as service, instance, or site goes up or down. The events are propagated by Oracle Notification System (ONS) or Streams Advanced Queuing (AQ). Compared to TCP/IP timeout, FAN provides fast detection of condition change and fast notification.

The FAN events can be used by the applications or users that connect to a new primary database upon failover by using Fast Connection Failover (FCF).

FCF

FCF is an Oracle High Availability feature for Java Database Connectivity (JDBC) applications and supports the JDBC Thin and JDBC Oracle Call Interface (OCI) drivers. FCF works with the JDBC connection caching mechanism, FAN, and Oracle RAC.

FCF provides the following High Availability features for client connections in planned and unplanned outages:

- Rapid database service, instance, or node failure detection then stops and removes invalid connections from the pool
- Recognition of new nodes that join an Oracle RAC cluster
- Load balancing the connection requests to all active Oracle RAC instances

Transparent Application Failover

Transparent Application Failover (TAF) is an OCI feature that provides the client recovery capabilities if connections fail. TAF can be used with or without FAN conditions:

- SELECT failover

  If the connection is lost, Oracle Net establishes a connection to another node and reruns the SELECT statements with the cursor positioned on the row on which it was positioned before the failover. This approach is best for data warehouse systems where the transactions are big and complex

- SESSION failover

  If a user's connection is lost, SESSION failover establishes a new session that is automatically created for the user on the backup node. This type of failover does not attempt to recover selects. This failover is ideal for Online Transaction Processing (OLTP) systems where transactions are small.

- Graceful session migration for planned downtime.

HADR applicability

A server failure, Linux crash, or other faults can cause the crash of an individual Oracle instance in an Oracle RAC database. To maintain availability, application clients that are connected to the failed instance are quickly notified of the failure and immediately established with a new connection to the surviving instances of the Oracle RAC database.

9.2.8 Oracle Data Guard technology

Oracle Data Guard configuration consists of one primary database and one or more (up to 30) standby databases. Oracle Data Guard maintains standby databases as transactionally consistent copies of the primary database. If the primary database becomes unavailable, Oracle Data Guard can switch any standby database to the primary role, which minimizes the downtime that is associated with the outage.

The following standby databases are available:

- Physical standby database
- Oracle Active Data Guard
- Transient logical standby database
- Snapshot standby database
- Logical standby database

An Oracle Data Guard configuration can include any combination of these types of standby databases.

Physical standby database

Physical standby databases include the following features:

- A physically identical copy of the primary database with identical schemas, indexes, and data files.
- Is kept synchronized with the primary database through Redo Apply, which recovers the redo data that is received from the primary database and applies the redo data to the physical standby database. This ensures a physical, block-for-block copy of the primary database.
- Physical standby database can be opened for read-only access while redo data is applied (Oracle Active Data Guard option or real-time query mode).

- Physical standby database can be used for taking backup, incremental backups, report creations, and creating clone databases.

Oracle Active Data Guard database

Oracle Active Guard databases include the following features:

- Is a superset of Data Guard and allows a physical standby database to be open read-only while changes are applied to it from the primary database.
- Enables productive use of physical standby databases
- Automatically repairs block corruptions that are detected at the primary database.

Transient Logical Standby databases

Current physical standby database can be temporarily converted to a logical standby and can be used for rolling database upgrades as recommended by Oracle MAA best practices.

Snapshot standby databases

Snapshot Standby databases include the following features:

- A physical standby database can be temporarily converted into a standby database that can be updated.
- Snapshot standby databases can be used as clones or test databases to validate new functionality and new releases. When finished, it can be converted back into a physical standby.
- While running in the snapshot standby database role, it continues to receive and queue redo data so that data protection and the RPO are maintained.
- When it is converted back to physical standby database, the changes that are made to the snapshot standby state are discarded. Redo Apply automatically resynchronizes the physical standby database with the primary database by using the redo data that was archived.

Logical standby databases

Logical standby databases include the following features:

- A logical standby database contains the same logical information as the primary database, although the physical organization and structure of the data can be different.
- The logical standby database is kept synchronized with the primary database through SQL Apply, which transforms the redo data that is received from the primary database into SQL statements and then runs the SQL statements on the standby database.

HADR applicability

HADR availability includes the following features:

- Data Guard technology addresses High Availability and Disaster Recovery requirements.
- Data Guard technology complements Oracle RAC.
- Provides one or more synchronized standby databases and protects data from failures, disasters, errors, and corruptions.

9.2.9 Oracle GoldenGate

Oracle GoldenGate is an asynchronous, log-based, real-time data replication technology that includes the following features:

- Moves data across heterogeneous database, hardware, and operating system environments.
- Supports multi-master replication, hub-and-spoke deployment, and data transformation.
- Supports replication that involves a heterogeneous mix of Oracle databases and non-Oracle databases.
- Can be deployed for data distribution and data integration.

HADR applicability

HADR applicability includes the following features:

- Maintains transactional integrity; it is resilient against interruptions and failures.
- Heterogeneous replication, transformations, subsetting, and multiple topologies.
- All sites fully active (read/write).

9.3 High Availability with z/VM

In a typical Oracle on Linux on System z environment, Linux can be implemented to run on a single LPAR or multiple Linux guests can be hosted in an LPAR that is running z/VM. Oracle databases are installed on the Linux guests.

In that environment, the unavailability or a single point of failures can be result of any of the following activities:

- Planned downtime activities:
  - System z hardware upgrades that require Power On Reset (POR).
  - LPAR configuration changes that require reboot of the LPAR.
  - z/VM maintenance.
- Unplanned outages:
  - The System z hardware might experience multiple unrecoverable failures, which cause the entire server to fail (although this is not likely to happen).
  - Network or connectivity failures.
  - Disk subsystem I/O channels failures.
  - The LPAR microcode might fail.
  - z/VM failures.

HADR applicability

z/VM offers the following technologies to enhance the High Availability of Oracle databases on Linux on System z environment:

- Single points of system failures can be avoided by implementing z/VM multi-system clustering technology, as described in Chapter 6, “Using z/VM Live Guest Relocation to relocate a Linux guest” on page 117.
- Ability to use multiple LPARs and distributing Linux guests to run on them reduces several potential single points of failure at the system-image level.

- The applications that are running on LPARs in a single System can communicate with each other by using HiperSockets or memory-to-memory data transfers. This avoids any external traffic and is a good choice to implement Oracle RAC interconnect requirements.
- Virtual switch (VSWITCH) under z/VM and OSA Channel Bonding under LPAR can be used to avoid network single point of failures.
- For ECKD DASD devices that are accessed over FICON® channels, redundant multipathing is provided and handled invisibly to the Linux operating system.
- For SCSI (fixed-block) LUNs that are accessed over System z FCP channels, each path to each LUN appears to the Linux operating system as a different device. Linux kernel (2.6 and above) multipath facility handles this and provides High Availability.

Tip: For more information about High Availability with System z, see *High-Availability of System Resources: Architectures for Linux on IBM System z Servers*, ZSW03236-USEN-01, which is available at this website:

http://public.dhe.ibm.com/common/ssi/ecm/en/zsw03236usen/ZSW03236USEN.PDF

9.4 Disaster Recovery solutions

One of the objectives in achieving High Availability is to prevent a site from becoming a single point of failure (SPoF). Disaster Recovery (DR) solutions are an extension of High Availability solutions with the added capability of providing resiliency with geographic dispersion. Current disaster recovery solutions require geographic dispersion and should also meet RTO, RPO, and SLA objectives.

A plan for DR normally includes the following considerations:

- Ensuring continuity of operations in the event of various disaster scenarios.
- *Dual site concept*, where two data centers are in different locations. The entire hardware configuration is redundant, and the two systems are connected to each other.
- Continuous operations for applications, databases, system, networks, supporting staff, and supporting infrastructure (power, cooling, and space).
- Normally DR processes and Business Continuity processes function as a closely coupled set of processes.
- DR configuration that is identical across tiers on the production site and standby site is called a *symmetric site*.
- Having a DR site is an expensive proposition because of the following costs:
  - Hardware
  - Software
  - Network
  - Site facilities
  - Human Resources
  - Under-used standby resources
  - No immediate ROI until a disaster occurs
- An ideal DR solution includes the following features:
  - Highly reliable
  - Low complexity
  - Proven technologies
  - Less expensive to implement

- Challenges in a DR solution include the following factors:
  - Expensive, redundant systems that are under-used
  - Difficult to test to determine whether it really works
  - No ROI until a disaster occurs
  - Hardware and software maintenance still needed
  - Distance across data centers creates data synchronization challenges

Many of the System z customers have well-established business processes for DR scenarios, which usually uses the Capacity BackUp (CBU) features of System z[1]. Their current DR environments also can be easily extended to include Oracle databases that are running on the Linux on System z environment. For Oracle databases, the major requirement for DR is data resiliency and can be achieved by any of the following technologies:

- Storage array-based remote mirroring solutions
- Extended cluster solutions (Extended RAC)
- Oracle Data Guard-based solutions.

Oracle MAA recommends that you build a DR solution that is based on Oracle Data Guard technology for Oracle databases for the following reasons:

- Automatic and fast failover
- Transactionally consistent data
- Detection and deletion of data corruptions
- Application, system vendor, or storage independent
- Planned downtime reduction by using database rolling upgrades

9.5 Summary

In this chapter, we described how to plan for a highly available and disaster recovery (HADR) environment for Oracle Databases that are running on Linux on System z in a virtualized environment.

HADR solutions are possible by using Oracle MAA blueprint with IBM System z hardware's proven design for continuous availability, which offers a set of reliability, availability, and serviceability features (RAS). The right HADR configuration is a balance between recovery time and recovery point requirements and cost.

[1] http://www-03.ibm.com/systems/z/advantages/resiliency/datadriven/cuod.html#temporary

Part 3

Provisioning an Oracle environment on Linux on System z

In this section, we describe several alternatives for provisioning a Linux guest for an Oracle Database on System z. The following methods are available:

- Provisioning an environment by using scripts (which are provided), as described in Chapter 10, “Automating Oracle on System z” on page 199.
- Provisioning an environment by using Tivoli Products, as described in Chapter 11, “Provisioning an Oracle environment” on page 231.
- Provisioning an environment by using Velocity software z/PRO, which is described in Chapter 12, “Using z/Pro as a Cloud infrastructure for Oracle” on page 279.

10

Automating Oracle on System z

Attention: The following command conventions are used in this chapter:

- z/VM commands are prefixed with ==>
- Linux commands that are running as root are prefixed with #
- Linux commands that are running as non-root are prefixed with $

The following general approaches can be used to automate the creation of virtual servers:

- Cloning: Copying an image of an existing Linux system
- Building: Installing a new Linux system with all configuration parameters supplied

Cloning is more commonly implemented because the golden image can be customized manually. Building is more systematic and reliable because every step to arrive at a cloned Linux is captured and can be audited. The cloning approach is described in this chapter.

Virtualization, or *cloud computing*, is often explained as the following three different levels:

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

IaaS can be considered the creation of virtual machines with no operating system installed or running. z/VM is an ideal hypervisor for this. PaaS can be considered adding an operating system to the infrastructure layer. Today, Linux is the most common OS to run under z/VM. SaaS can be considered the installation and configuration of software to the platform layer. This chapter describes adding Oracle software.

This chapter was written by using Red Hat Enterprise Linux (RHEL) 6.2 as the reference distribution. The bash scripts that are associated with this book might work on other versions or RHEL but does not work on any version of SUSE Linux Enterprise Server. Ideally, both scripts should work on SUSE Linux Enterprise Server and Red Hat Enterprise Linux. However, both distributions are not described here.

This chapter includes of the following topics:

- Infrastructure as a Service under z/VM
- PaaS
- SaaS for Oracle stand-alone

10.1 Infrastructure as a Service under z/VM

To get started, the following System z resources must be obtained and configured:

- At least one System z Logical Partition (LPAR)
- z/VM installed on the LPAR and configured
- Processors (CPUs): More commonly IFLs or possibly CPs
- Memory: Defined to the LPAR as central and expanded
- Disk space: ECKD DASD and SCSI/FCP LUNs
- FCP devices: Accesses the SCSI/FCP LUNs
- TCP/IP addresses
- OSA devices to access the network

In the environment that is described in this chapter, z/VM 6.2 at the latest service level (1201) was installed on an LPAR and sufficient processors (four), memory (26 GB central, 2 GB expanded), disk space, and networking resources (OSA devices and TCP/IP addresses) were available.

Complete the tasks that are described in this section to implement IaaS.

10.1.1 Configuring z/VM

The following z/VM configurations are made:

- The `SYSTEM CONFIG` file is configured.
- TCP/IP is set up and the FTP server is turned on.
- The directory manager DirMaint is configured.
- The Systems Management APIs (SMAPI) are configured.

For the `SYSTEM CONFIG` file configuration, some details are provided in this section. For more information about TCP/IP and the FTP server, see “Configure TCP/IP” and “Turn on the z/VM FTP server and on DirMaint and SMAPI” in Chapter 18 of *The Virtualization Cookbook for z/VM 6.2 RHEL 6.2 and SLES 11 SP2*, which is available at this website:

http://www.vm.ibm.com/devpages/mikemac/CKB-VM62.PDF

The `SYSTEM CONFIG` file is the first configuration file that z/VM processes when it loads initially. A layer 3 virtual switch named `VSWITCH1` is created for the z/VM TCP/IP stack. A layer 2 virtual switch named `VSWITCH2` is created for the Linux systems’ primary network interfaces. Oracle Grid requires a private interconnect among all nodes in a cluster. A third layer 2 virtual switch named `VSWITCH3` with no OSA connection is created for this purpose. Layer 2 virtual switches are now recommended over layer 3 because they are required for DHCP and IPv6.

To customize the `SYSTEM CONFIG` file, complete the following steps:

1. Link and access the `PMAINT CF0` disk, as shown in the following example:

```
==> link pmaint cf0 cf0 mr
==> acc cf0 f
```

2. Edit the `SYSTEM CONFIG` file, as shown in the following example:

```
==> x system config f
```

3. The following highlighted items are configured in this file. At the top, the many DASD volumes are added as **User_Volume_List** statements so that z/VM can use them as minidisks:

```
/**********************************************************************/
/* User volumes for local minidisks                                   */
/**********************************************************************/

               User_Volume_List  LX7W01 LX7U1R
               User_Volume_List  LX6601
               User_Volume_List  LX6602
               ...
```

4. Add the **Disconnect_Timeout off** and **Vdisk** clauses in the **Features** statement and configure the system so that disconnected users are not forced off, and Linux virtual machines can create virtual disks for in-memory swap spaces, as shown in the following example:

```
               ...                              # many more volumes
Features ,
   Disable ,                      /* Disable the following features */
     Set_Privclass ,              /* Disallow SET PRIVCLASS command */
     Auto_Warm_IPL ,              /* Prompt at IPL always           */
     Clear_TDisk   ,              /* Don't clear TDisks at IPL time */
   Enable,
     STP_TZ,                      /* timezone from STP on CPC       */
   Retrieve ,                     /* Retrieve options               */
     Default  99 ,                /* Default.... default is 20      */
     Maximum  255 ,               /* Maximum.... default is 255     */
   MaxUsers noLimit ,             /* No limit on number of users    */
   Passwords_on_Cmds ,            /* What commands allow passwords? */
     Autolog  yes ,               /* ... AUTOLOG does               */
     Link     yes ,               /* ... LINK does                  */
     Logon    yes ,               /* ... and LOGON does, too        */
   Disconnect_Timeout off ,       /* Don't force disconnected users */
   Vdisk ,                        /* Allow VDISKS for Linux swaps   */
     Syslim infinite ,
     Userlim infinite
...
```

5. Define virtual switches. In this example, devices 2040-2045 are OSA devices on one CHPID and OSA card, while devices 2060-2065 are on a different CHPID on another OSA card. This architecture prevents VSWITCH1 and VSWITCH2 from having a single point of failure, as shown in the following example:

```
/**********************************************************************/
/*                       VSWITCHes                                    */
/* VSWITCH1 - layer 3 - z/VM TCPIP stack                              */
/* VSWITCH2 - layer 2 - Linux primary interfaces                      */
/* VSWITCH3 - layer 2 - Linux secondary interconnect - no OSA         */
/*                                                                    */
/**********************************************************************/

 DEFINE VSWITCH VSWITCH1 RDEV 2040 2060
 MODIFY VSWITCH VSWITCH1 GRANT TCPIP
 DEFINE VSWITCH VSWITCH2 ETHERNET RDEV 2043 2063
 DEFINE VSWITCH VSWITCH3 ETHERNET
```

The `SHUTDOWN` time is set to 10 minutes (600 seconds). With this setting, when z/VM is shut down, it sends a signal to all guests and each Linux system has up to 10 minutes to shut down cleanly, as shown in the following example:

```
/*********************************************************************/
/*                      Set signal for shutdown                      */
/*********************************************************************/

SET SIGNAL SHUTDOWN 600
...
```

z/VM should now be sufficiently configured to install Oracle.

10.1.2 Defining virtual machines

A profile and virtual machines are defined in z/VM by using the user directory. In the reference environment, DirMaint is configured so the **DIRM ADD** command is used to create the profile and virtual machines.

Complete the tasks that are described in this section.

Defining a virtual machine for a common 191 disk

Define a virtual machine (named `LNXMAINT` in this example) to provide for a common read-only `191` disk for all Linux virtual machines. The label **JM6289** is shown in bold in the following example (and other examples in this chapter) because that shows values that must be changed for your enterprise:

```
USER LNXMAINT LNXMAINT 64M 128M BEG
INCLUDE TCPCMSU
LINK TCPMAINT 592 592 RR
MDISK 0191 3390 0001 0020 JM6289 MR READ WRITE MULTIPLE
MDISK 0192 3390 0021 0500 JM6289 MR ALL WRITE MULTIPLE
```

The `191` disk is only used for this virtual machine's `PROFILE EXEC`. The `192` disk, which is read-write, becomes the read-only `191` disk to all other Linux virtual machines. This is where the installation files (kernel and RAMdisk) and the parameter and configuration files are kept.

Defining a profile for all Linux virtual machines

A z/VM user directory profile allows statements that are common to a similar set of virtual machines to be shared. As shown in the following example, the profile named `LNXDFLT` is common to all Linux systems. The first six command statements allow access to virtual switches `VSWITCH2` and `VSWITCH3`. All Linux virtual machines that use this profile have two network interface controllers (NICs) to the public interface and the primary interconnect:

```
PROFILE LNXDFLT
   COMMAND SET VSWITCH VSWITCH2 GRANT &USERID
   COMMAND DEFINE NIC 600 TYPE QDIO
   COMMAND COUPLE 600 TO SYSTEM VSWITCH2
   COMMAND SET VSWITCH VSWITCH3 GRANT &USERID
   COMMAND DEFINE NIC 700 TYPE QDIO
   COMMAND COUPLE 700 TO SYSTEM VSWITCH3
   CPU 00 BASE
   CPU 01
   IPL CMS
   MACHINE ESA 8
   OPTION CHPIDV ONE
```

```
CONSOLE 0009 3215 T
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
LINK LNXMAINT 0192 0191 RR
LINK TCPMAINT 0592 0592 RR
LINK ZPRO 0391 0391 RR
```

The last LINK statement is required if you are using Velocity Software's zPro.

Defining virtual machines for golden images

The following example shows a virtual machine that is defined for the RHEL 6.2 golden image. The `100` disk is where Linux is installed. The `101` disk becomes a logical volume for the Oracle binaries. After two VDISKs are created at `300` and `301` at virtual machine creation time, the `302` disk becomes the third, slow, swap space:

```
USER RH62GOLD ORACLE 1G 6G G
   INCLUDE LNXDFLT
   MDISK 0100 3390 1 10016 LX9A12 MR
   MDISK 0101 3390 1 30050 LX6602 MR
   MDISK 0302 3390 1 10016 LX9A13 MR
```

A similar virtual machine, `S112GOLD`, is created for the SUSE Linux Enterprise Server 11 SP2 golden image.

Defining virtual machines for stand-alone Oracle servers

The following example shows a virtual machine that is defined for a stand-alone of Single Image (SI) Oracle system. The disk device numbers and sizes are identical to the golden images. The two `DEDICATE` statements of real devices `400` and `500` allows multipathing to FCP/SCSI LUNs:

```
USER LNXSA1 ORACLE 1G 6G G
  INCLUDE LNXDFLT
  MDISK 0100 3390 0001 10016 LX9A1A MR
  MDISK 0101 3390 0001 30050 LX6605 MR
  MDISK 0302 3390 0001 10016 LX9A1B MR
  DEDICATE 0400 B800
  DEDICATE 0500 B900
```

The virtual machines `LNXSA2` and `LNXSA3` are also defined for a reference environment.

Defining virtual machines for Oracle clusters

The following example shows that two virtual machines are defined for a minimal "1.1" Oracle cluster:

- Define the first virtual machine for the cluster with real minidisks for Oracle data disks. Minidisks `200-202` are used for voting disks and `210-213` are used for Oracle data and archives. The `MDISK` modes of `MW` for the ASM disks allow multiple-write access. The `V` mode suffix tells CP to use its virtual reserve/release support, which Oracle ASM uses. The `MINIOPT NOMDC` statements turn off minidisk cache for the minidisks that precede them.

The DEDCIATE 400 and 500 supply FCP devices for access to the SAN, as shown in the following example:

```
USER LNXC2N1 ORACLE 4G 6G G
   INCLUDE LNXDFLT
   MDISK 0100 3390 0001 10016 LX9A1D MR
   MDISK 0101 3390 0001 30050 LX6606 MR
   MDISK 0200 3390 1 1000 LX9A0E MR
   MINIOPT NOMDC
   MDISK 0201 3390 1001 1000 LX9A0E MR
   MINIOPT NOMDC
   MDISK 0202 3390 2001 1000 LX9A0E MR
   MINIOPT NOMDC
   MDISK 0302 3390 20033 10016 LX6705 MR
   MINIOPT NOMDC
   DEDICATE 0400 B803
   DEDICATE 0500 B903
```

- Define other virtual machines to be part of the cluster. There are LINK statements to the seven Oracle ASM disks on the first node in the cluster in MW mode. Normally, this link mode is dangerous; however, Oracle CRS is designed to work with multiple systems that are writing to the same disks, as shown in the following example:

```
USER LNXC2N2 ORACLE 4G 6G G
   INCLUDE LNXDFLT
   MDISK 0100 3390 0001 10016 LX9A0A MR
   MDISK 0101 3390 0001 30050 LX6702 MR
   LINK LNXC1N1 0200 0200 MW
   LINK LNXC1N1 0201 0201 MW
   LINK LNXC1N1 0202 0202 MW
   MDISK 0302 3390 0001 10016 LX9A0B MR
   DEDICATE 0400 B804
   DEDICATE 0500 B904
```

This completes the profile and virtual machine definitions for a reference environment. This configuration can be considered Iaas.

10.2 PaaS

Now that a virtual machine is defined, an operating system can be added to it. This can be considered PaaS. A Linux distribution must be obtained. This chapter focuses on Red Hat Enterprise Linux (RHEL) 6.2, which is now formally certified by Oracle.

To set up PaaS, complete the tasks that are described in this section.

10.2.1 Preparing to install Red Hat Enterprise Linux 6.2 on the golden image

When RHEL 6.2 Linux is installed, two CMS files are needed: a parameter file and a configuration file. The parameter file has a few common settings and it points to the configuration file. It is common to keep all Linux parameter and configuration files on the same disk, with each Linux virtual machine linking to them read-only. In this example, this is the LNXMAINT 192 disk.

Create the parameter and configuration files on the `LNXMAINT 192` disk with the user ID being the file name. This disk is the read-only `191` disk for all Linux virtual machines. Following are the contents of those two files:

```
==> type rh62gold parm-rh6 d

root=/dev/ram0 ro ip=off ramdisk_size=40000
CMSDASD=191 CMSCONFFILE=RH62GOLD.CONF-RH6
vnc vncpassword=12345678

==> type rh62gold conf-rh6 d

DASD=100-101,300-302
HOSTNAME=rh62gold.itso.ibm.com
NETTYPE=qeth
IPADDR=9.12.7.2
SUBCHANNELS=0.0.0600,0.0.0601,0.0.0602
NETMASK=255.255.240.0
SEARCHDNS=itso.ibm.com
GATEWAY=9.12.4.1
DNS=9.12.6.7
MTU=1500
PORTNAME=DONTCARE
PORTNO=0
LAYER2=1
IPADDR2=10.1.1.2
```

All variables are recognized by the RHEL installer except for `IPADDR2`. This variable is used later by the first boot script, `boot.onetime`, for the private interconnect.

10.2.2 Installing Red Hat Enterprise Linux 6.2 Linux on the golden image

RHEL 6.2 is installed onto the `RH62GOLD` virtual machine (a detailed description of the installation is outside the scope of this book). For more information, see Chapter 8 of *The Virtualization Cookbook for z/VM 6.2 RHEL 6.2 and SLES 11 SP2*, which is available at this website:

http://www.vm.ibm.com/devpages/mikemac/CKB-VM62.PDF

The virtual machines are given a 3390-9 (approximately 7 GB) for the Linux system, a 3390-27 (approximately 21 GB) for the Oracle binaries, and another 3390-9 for a swap space on disk.

The root file system is not put into a logical volume, but many other file systems are. The rationale for this is that logical volumes allow file system sizes to be easily extended. The root file system is in a single partition because if there is any problem with logical volumes, the Linux system still boots. Because the root file system cannot grow and is relatively small (512 MB), it is important that it is not allowed to fill up. Data of large sizes should be put into logical volumes such as `/tmp/` or `/opt/`.

The FCP/SCSI LUNs for Oracle data are not added during the Linux installation. This is done after cloning is completed.

Table 10-1 shows a summary of the file systems and volumes.

Table 10-1 File system layout for RHEL 6.2

Mount point	Size	Volume group name	Logical volume name	Minidisk
/	512 MB	None	None	100
/tmp/	1 GB	system_vg	tmp_lv	100
/usr/	3 GB	system_vg	usr_lv	100
/var/	512 MB	system_vg	var_lv	100
/opt/	20 GB	opt_vg	opt_lv	101
swap	7 GB	None	None	302

After starting the RHEL 6.2 install, but before starting an SSH session as the user `install`, there is a possible intermediate step that is necessary. There is an issue in which the Red Hat installer does not recognize disks that were formatted with `CPFMTXA`. If you used the **dasdfmt** command to format the minidisks, you can skip this step. If not, start an SSH session and use the **dasdfmt** command to format the disks. Complete the following steps:

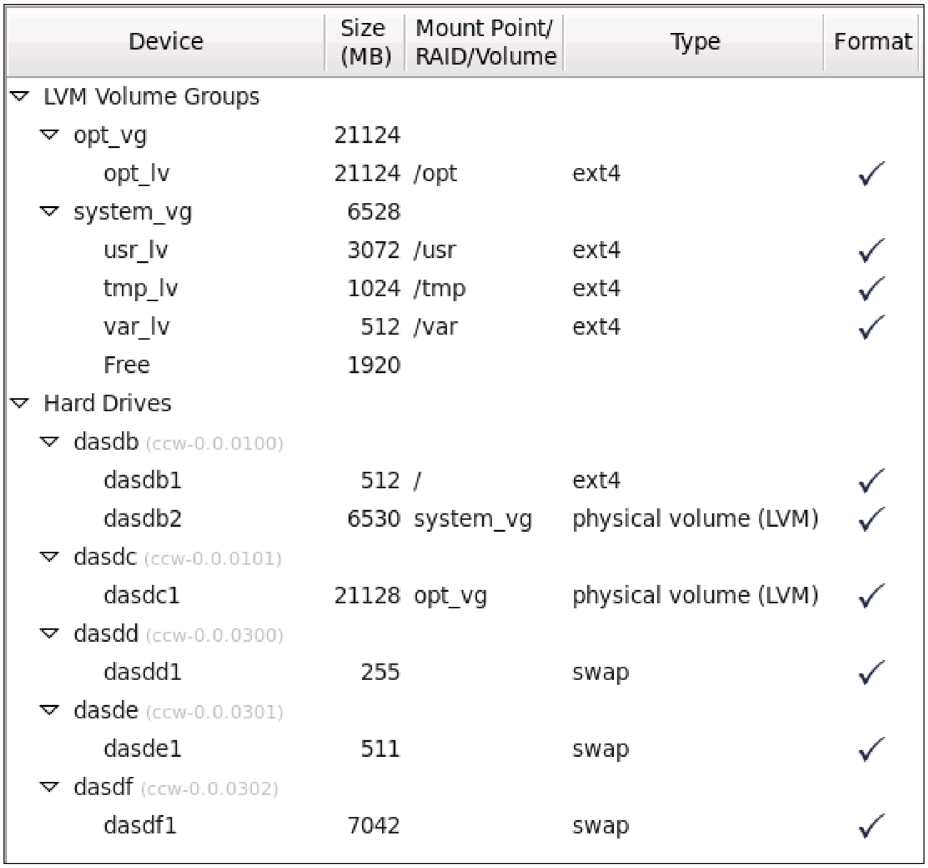
1. Start SSH session to the all system and log in as `root`. A password is not required.
2. Run the **lsdasd** command to observe the disks. In the following example, `dasdb`, `dasdc`, and `dasdf` are minidisks that must be formatted:

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasdb     94:4    ECKD  ???    7042MB    ???
0.0.0101   active      dasdc     94:8    ECKD  ???    21128MB   ???
0.0.0300   active      dasdd     94:12   FBA   ???    256MB     ???
0.0.0301   active      dasde     94:16   FBA   ???    512MB     ???
0.0.0302   active      dasdf     94:20   ECKD  ???    7042MB    ???
```

3. Use a bash **for** loop and put the **dasdfmt** commands in the background so the formats can be performed in parallel, as shown in the following example:

```
# for i in b c f
> do
>   dasdfmt -b 4096 -y -f /dev/dasd$i &
> done
[1] 640
[2] 641
[3] 642
```

4. When you are prompted by the question from the installer: “Which type of installation would you like?”, select **Create Custom Layout**. Use the GUI tools to create file systems, swap spaces, and logical volumes. As shown in Figure 10-1, the Summary window shows the disk and swap space layout.



Device	Size (MB)	Mount Point/ RAID/Volume	Type	Format
LVM Volume Groups				
opt_vg	21124			
opt_lv	21124	/opt	ext4	✓
system_vg	6528			
usr_lv	3072	/usr	ext4	✓
tmp_lv	1024	/tmp	ext4	✓
var_lv	512	/var	ext4	✓
Free	1920			
Hard Drives				
dasdb (ccw-0.0.0100)				
dasdb1	512	/	ext4	✓
dasdb2	6530	system_vg	physical volume (LVM)	✓
dasdc (ccw-0.0.0101)				
dasdc1	21128	opt_vg	physical volume (LVM)	✓
dasdd (ccw-0.0.0300)				
dasdd1	255		swap	✓
dasde (ccw-0.0.0301)				
dasde1	511		swap	✓
dasdf (ccw-0.0.0302)				
dasdf1	7042		swap	✓

Figure 10-1 File system and swap space layout in RHEL 6.2 golden image

5. Complete the following steps to choose the software during the installation process:
   a. Click **Customize now** at the bottom of the main panel, as shown in the top of Figure 10-2 on page 209.
   b. Remove some package groups that were deleted from the Base System group, as shown in Figure 10-2 on page 209.
   c. Add two package groups to the Development group. The development tools were added because many software products require access to GNU Collection of Compilers (`gcc`) and the associated tools and libraries that it pulls in.

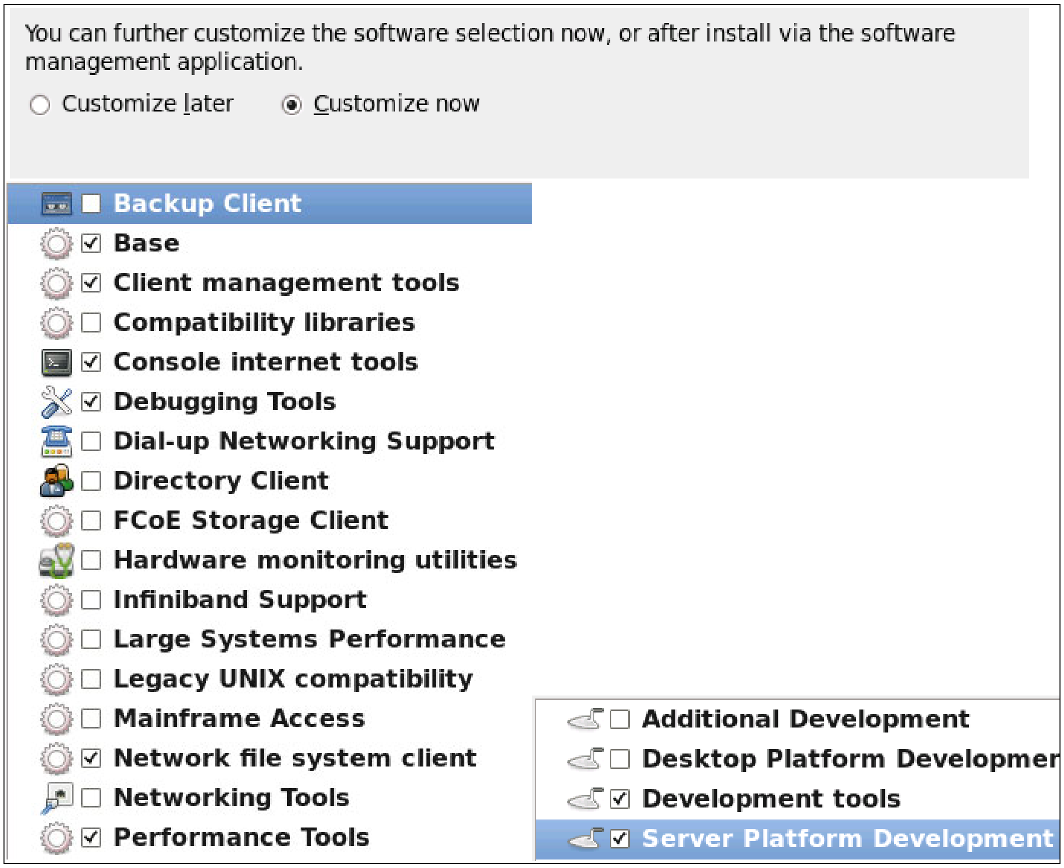


Figure 10-2 Software installation choices in Base System and Development groups

After the installation is complete, start an SSH session and observe the file systems and swap spaces with the **df -h** and **swapon -s** commands. The reference system is shown in the following example:

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasdd1           504M  162M  317M  34% /
tmpfs                 498M     0  498M   0% /dev/shm
/dev/mapper/opt_vg-opt_lv
                       21G  172M   20G   1% /opt
/dev/mapper/system_vg-tmp_lv
                     1008M   34M  924M   4% /tmp
/dev/mapper/system_vg-usr_lv
                      3.0G  1.5G  1.4G  53% /usr
/dev/mapper/system_vg-var_lv
                      504M   62M  417M  13% /var
# swapon -s
Filename                                Type            Size    Used
Priority
/dev/dasdc1                             partition       259956  0       -1
/dev/dasdb1                             partition       519924  0       -2
/dev/dasda1                             partition       7211416 0       -3
```

The RHEL 6.2 golden image should now be installed and is ready to be configured.

10.2.3 Configuring the Red Hat Enterprise Linux 6.2 golden image

The tasks that are described in this section are recommended to configure the golden image. While some of these tasks might seem specific to Oracle servers, often they are helpful for all Linux systems regardless of the running workload.

Adding a network interface to the private interconnect

Linux was installed onto a single network interface by using the NIC starting at virtual device address 600. Oracle Grid requires a private interconnect, but having a second interface for all Linux systems in the LPAR to communicate over can be useful.

To add a second interface for the private interconnect, complete the following steps:

1. Show the NICs that are defined to the virtual machine by using the **CP QUERY NIC** command that is run with the **vmcp** module, as shown in the following example:

```
# modprobe vmcp
# vmcp q nic
Adapter 0600.P00 Type: QDIO      Name: DONTCARE    Devices: 3
  MAC: 02-00-00-00-00-07         VSWITCH: SYSTEM VSWITCH2
Adapter 0700.P00 Type: QDIO      Name: UNASSIGNED  Devices: 3
  MAC: 02-00-00-00-00-08         VSWITCH: SYSTEM VSWITCH3
```

   There is also a second NIC starting at virtual device address 700. This was created in the profile LNXDFLT and it is attached to VSWITCH3.

2. List the configured NIC by using the **znetconf -c** command, as shown in the following example:

```
# znetconf -c
Device IDs                 Type    Card Type      CHPID Drv. Name        State
-------------------------------------------------------------------------------
0.0.0600,0.0.0601,0.0.0602 1731/01 GuestLAN QDIO     00 qeth eth0        online
```

   This shows that the 600 NIC is configured and associated with the eth0 interface.

3. List the unconfigured NIC by using the **znetconf -u** command, as shown in the following example:

```
# znetconf -u
Scanning for network devices...
Device IDs                 Type    Card Type      CHPID Drv.
------------------------------------------------------------
0.0.0700,0.0.0701,0.0.0702 1731/01 OSA (QDIO)        01 qeth
```

   This shows that the 700 NIC is not yet configured.

4. Configure all NICs by using the **znetconf -A** command and again view the configured NICs, as shown in the following example:

```
# znetconf -A
Scanning for network devices...
Successfully configured device 0.0.0700 (eth1)
# znetconf -c
Device IDs                 Type    Card Type      CHPID Drv. Name        State
-------------------------------------------------------------------------------
-
0.0.0600,0.0.0601,0.0.0602 1731/01 GuestLAN QDIO     00 qeth eth0        online
0.0.0700,0.0.0701,0.0.0702 1731/01 GuestLAN QDIO     01 qeth eth1        online
```

   This shows that the 700 NIC is now configured and associated with eth1.

5. View the /etc/sysconfig/network-scripts/ifcfg-eth0 file, which is the primary network interface configuration file, as shown in the following example:

```
# cd /etc/sysconfig/network-scripts/
# cat ifcfg-eth0
DEVICE="eth0"
BOOTPROTO="static"
DNS1="9.12.6.7"
DOMAIN="itso.ibm.com"
GATEWAY="9.12.4.1"
IPADDR="9.12.7.2”
MTU="1500"
NETMASK="255.255.240.0"
NETTYPE="qeth"
NM_CONTROLLED="yes"
ONBOOT="yes"
OPTIONS="layer2=1 portno=0"
PORTNAME="DONTCARE"
SUBCHANNELS="0.0.0600,0.0.0601,0.0.0602"
TYPE="Ethernet"
```

6. To add another interface, some of the following values must be changed:
   - DEVICE: Interface name
   - IPADDR: IP address
   - SUBCHANNELS: Virtual device addresses of the NIC coupled to the z/VM VSWITCH

   Copy the configuration file for eth0 to eth1 and modify the DEVICE, GATEWAY, IPADDR, and NETMASK (optional) variables, depending on your network setup, as shown in the following example:

```
# cp ifcfg-eth0 ifcfg-eth1
# vi ifcfg-eth1
DEVICE="eth1"
BOOTPROTO="static"
DNS1="9.12.6.7"
DOMAIN="itso.ibm.com"
GATEWAY=""
IPADDR="10.1.1.2"
MTU="1500"
NETMASK="255.255.255.0"
NETTYPE="qeth"
NM_CONTROLLED="yes"
ONBOOT="yes"
OPTIONS="layer2=1 portno=0"
PORTNAME="DONTCARE"
SUBCHANNELS="0.0.0700,0.0.0701,0.0.0702"
TYPE="Ethernet"
```

   Because there is no route off the network, the GATEWAY variable is set to the null string. The Subnetting of the interconnect is set to a class C (or /24), so the subnet mask is changed to 255.255.255.0.

7. Reboot the system, as shown in the following example:

```
# reboot

Broadcast message from root@rh62gold.itso.ibm.com
        (/dev/pts/0) at 7:58 ...
```

8. Start a new SSH session as `root` and run the **`ifconfig eth1`** command. You should see the new network interface, as shown in the following example:

```
The system is going down for reboot NOW!
...
# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 02:00:00:00:00:08
          inet addr:10.1.1.2  Bcast:10.1.1.255  Mask:255.255.255.0
          inet6 addr: fe80::ff:fe00:8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:830 (830.0 b)
```

If this interface is not present or does not show an IP address, you must remedy the problem.

Configuring the yum command on the RHEL 6.2 golden image

Software packages can always be added by using the **`rpm`** command. They also can be added to all co-requisite packages recursively by using the **`yum`** command. It must be configured to point to the all source (in this example, the FTP server at IP address 9.12.4.69, user name is `totibm`, and the password is `password`).

Complete the following steps:

1. Change directory to `/etc/yum.repos.d/`, as shown in the following example:

```
# cd /etc/yum.repos.d/
```

2. Create the file named `rhel62.repo`. Include in it the URL from which RHEL 6.2 was installed, as shown in the following example:

```
# vi rhel62.repo
[RHEL62]
name=Red Hat Enterprise Linux 6.2
baseurl=ftp://totibm:password@9.12.4.69//code/rhel62-s390x/dvd1
```

3. Copy the GPG KEY file for the RHEL 6.2 release, as shown in the following example:

```
# scp totibm@9.12.4.69:/code/rhel62-s390x/dvd1/RPM-GPG-KEY-redhat-release .
The authenticity of host '9.12.4.69 (9.12.4.69)' can't be established.
RSA key fingerprint is 01:be:16:9d:5d:b6:8a:e0:77:87:ef:a5:f8:37:2b:c3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '9.12.4.69' (RSA) to the list of known hosts.
Password:
RPM-GPG-KEY-redhat-release                        100% 3211     3.1KB/s   00:00
```

4. Install it by using the **`rpm --import`** command, as shown in the following example:

```
# rpm --import RPM-GPG-KEY-redhat-release
```

5. Test the new configuration by using the **`yum grouplist`** command, as shown in the following example:

```
# yum grouplist
Loaded plugins: product-id, security, subscription-manager
Updating certificate-based repositories.
Setting up Group Process
installed Groups:
   Additional Development
   Base
```

You should see a long list of package groups that was obtained from the installation server. If you do not see a long list of commands, you must remedy the problem.

Preparing for multipathing

The `device-mapper-multipath` RPM package is necessary for setting up multipathing to FCP/SCSI disks. Install it by using the **`yum -y all`** command, as shown in the following example:

```
# yum -y all device-mapper-multipath
...

================================================================================
 Package                          Arch      Version           Repository  Size
================================================================================
Installing:
 device-mapper-multipath          s390x     0.4.9-46.el6      RHEL62      92 k
Installing for dependencies:
 device-mapper-multipath-libs     s390x     0.4.9-46.el6      RHEL62     142 k

Transaction Summary
================================================================================
all       2 Package(s)

Total download size: 233 k
Installed size: 746 k
Downloading Packages:
(1/2): device-mapper-multipath-0.4.9-46.el6.s390x.rpm    |  92 kB     00:00
(2/2): device-mapper-multipath-libs-0.4.9-46.el6.s390x.r | 142 kB     00:00
--------------------------------------------------------------------------------
Total                                          172 kB/s | 233 kB     00:01
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : device-mapper-multipath-libs-0.4.9-46.el6.s390x            1/2
```

6. Create the file `/etc/multipath.conf`. This turns on *friendly names* (for example, `mpatha` instead of long names) and adds all ECKD DASD devices to the black list, so they are not considered for multipathing, as shown in the following example:

```
# cd /etc
# vi multipath.conf
##
## This is a template multipath-tools configuration file
## Uncomment the lines relevent to your environment
##
defaults {
  user_friendly_names yes
}
blacklist_exceptions {
  devnode "dasd[a-z]*"
}
```

The basic configuration of multipathing should now be configured.

Installing and configuring the VNC server

A typical Oracle installation requires a graphical environment. Linux graphical environments often are implemented with the X Window System or Virtual Network Computing (VNC), which is described in this section. The `tightvnc` package is a VNC server and allows for a graphical environment to be set up easily by using the **vncserver** command.

To set up a graphical environment with VNC, complete the following steps:

1. Install the VNC server and associated packages by using the following **yum** command:

```
# yum -y all tigervnc-server openmotif xterm xsetroot xorg-x11-xauth
...
Installed:
  openmotif.s390x 0:2.3.3-4.el6
  tigervnc-server.s390x 0:1.0.90-0.15.20110314svn4359.el6_1.1
  xorg-x11-server-utils.s390x 0:7.5-5.2.el6
  xorg-x11-xauth.s390x 1:1.0.2-7.1.el6
  xterm.s390x 0:253-1.el6

Dependency installed:
  libXaw.s390x 0:1.0.6-4.1.el6           libXdmcp.s390x 0:1.0.3-1.el6
  libXmu.s390x 0:1.0.5-1.el6             libXp.s390x 0:1.0.0-15.1.el6
  libXpm.s390x 0:3.5.8-2.el6             libXxf86misc.s390x 0:1.0.2-1.el6
  libmcpp.s390x 0:2.7.2-4.1.el6          libxkbfile.s390x 0:1.0.6-1.1.el6
  mcpp.s390x 0:2.7.2-4.1.el6             xkeyboard-config.noarch 0:2.3-1.el6
  xorg-x11-fonts-misc.noarch 0:7.2-9.1.el6 xorg-x11-xkb-utils.s390x 0:7.4-6.el6

Complete!
```

2. The VNC server configuration file is `/etc/sysconfig/vncservers`. Edit the file by adding one line at the bottom (another line is commented that can be used after the `oracle` and `grid` users are added), as shown in the following example:

```
# cd /etc/sysconfig
# vi vncservers
...
# VNCSERVERS="2:myusername"
# VNCSERVERARGS[2]="-geometry 800x600 -nolisten tcp -localhost"
# VNCSERVERS="1:root 2:oracle 3:grid"
VNCSERVERS="1:root"
```

   This setting allows `root` to start VNC session 1 (a comment is added if later the `oracle` user is to start session 2 and the `grid` user is to start session 3).

3. Set a VNC password by using the **vncpasswd** command. This password is needed to connect to the VNC server, as shown in the following example:

```
# vncpasswd
Password: 123456768
Verify: 12345678
```

4. Start the VNC server. This creates some initial configuration files under the `/root/.vnc/` directory, as shown in the following example:

```
# service vncserver start
Starting VNC server: 1:root
New 'rh62gold.itso.ibm.com:1 (root)' desktop is rh62gold.itso.ibm.com:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/rh62gold.itso.ibm.com:1.log

[  OK  ]
```

5. The directory /root/.vnc/ is where configuration files are kept. Change to that directory and list the files, as shown in the following example:

```
# cd /root/.vnc
# ls
passwd  rh62gold.itso.ibm.com:1.log  rh62gold.itso.ibm.com:1.pid  xstartup
```

6. The file xstartup is the script that is run when the VNC server starts and where the window manager is set. It is recommended that you change from the Tiny window manger, **twm**, to the more usable Motif window manager, **mwm**, as shown in the following example

```
# vi xstartup   // change last line
...
xsetroot -solid gray
vncconfig -iconic &
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
mwm &
```

7. Verify that the Motif window manager is available by using the **which** command, as shown in the following example:

```
# which mwm
/usr/bin/mwm
```

8. Restart the VNC server by using the **service** command, as shown in the following example:

```
# service vncserver restart
Shutting down VNC server: 1:root [  OK  ]
Starting VNC server: 1:root
New 'rh62gold.itso.ibm.com:1 (root)' desktop is rh62gold.itso.ibm.com:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/rh62gold.itso.ibm.com:1.log

[  OK  ]
```

You can now use the VNC client to connect to the IP address of the Linux administration system with a :1 appended. Supply the root password to start a session.

Copying files to Linux

Additional files are needed on the golden image to complete all the steps in this chapter. There are two boot scripts that are described in this chapter. The Web material that is associated with this book is available at this website:

ftp://www.redbooks.ibm.com/redbooks/SG248104

Also, there are two RPMs from Oracle that are necessary and one from Velocity Software if you plan to use any Velocity tools.

In this example, the .tar file that is associated with this Redbooks publication is copied to the /tmp/ directory and the **tar xzvf** command is used to extract the contents, as shown in the following example:

```
# cd /tmp
# tar xzvf SG248104.tgz
oracleRedbook-SG248104/
oracleRedbook-SG248104/README.txt
oracleRedbook-SG248104/disclaimer.txt
oracleRedbook-SG248104/vm/
oracleRedbook-SG248104/vm/clone.exec
oracleRedbook-SG248104/linux/
```

```
oracleRedbook-SG248104/linux/boot.oracle
oracleRedbook-SG248104/linux/boot.onetime
oracleRedbook-SG248104/linux/database.rsp
```

This creates one directory, `oracleRedbook-SG248104/`, with subdirectories `linux/` for Linux files and `vm/` for z/VM files.

Copy the two Linux boot scripts and the two response files to the `/etc/init.d/` directory on the RHEL 6.2 golden image. Also, copy the other files that are necessary for Oracle and Velocity Software, if applicable, as shown in Table 10-2.

Table 10-2 Files that are needed on golden image

File (RHEL 6.2)	Source	Description
`/etc/init.d/boot.onetime`	`SG248104.tgz` file	Configure network
`/etc/init.d/boot.oracle`	`SG248104.tgz` file	Prepare for Oracle all
`/tmp/database.rsp`	`SG248104.tgz` file	Database response file
`/tmp/grid.rsp`	`SG248104.tgz` file	Grid response file
`/tmp/LV11R6.s390x.rpm`	Velocity Software	New snmp MIBs
`/tmp/ora-val-rpm-EL6-DB-11.2.0.3-1.s390x.rpm`	Oracle	Pre-requisite checker
`/tmp/cvuqdisk-1.0.9-1.rpm`	Oracle	For Oracle Grid

You should now have all the additional files on the golden image.

Customizing for Velocity Software products

If you plan to use Velocity Software products, this step is necessary. In the reference environment, Velocity Pro was implemented. An RPM was installed to instruct `snmp` to add the Velocity Software MIB so that detailed Linux performance information can be retrieved by the software.

Complete the following steps:

1. Confirm that the file `LV11R6.s390x.rpm` was copied to the RHEL 6.2 golden image.
2. Show the file, then install it by using the **`rpm -ivh`** command, as shown in the following example:

```
# cd /tmp
# ls -l LV11R6.s390x.rpm
-rw-r--r--. 1 root root 22092 Oct 10 10:28 LV11R6.s390x.rpm
# rpm -ivh LV11R6.s390x.rpm
Preparing...                ########################################### [100%]
   1:libvsi                 ########################################### [100%]
```

3. Create a backup copy of the `snmpd.conf` file in the `/etc/snmp/` directory, as shown in the following example:

```
# cd /etc/snmp
# cp snmpd.conf snmpd.conf.orig
```

4. Rename the two `snmp*.conf.vsi` configuration files to `snmpd*.conf`, as shown in the following example:

```
# mv snmpd.conf.vsi snmpd.conf
# mv snmp.conf.vsi snmp.conf
```

5. Add the flag **-10** to the **/etc/init.d/snmpd** service script, as shown in the following example:

```
# cd /etc/init.d
# vi snmpd
...
# diff snmpd /tmp/snmpd.orig
45c45
<                 daemon -10 --pidfile=$pidfile $binary $OPTIONS
---
>                 daemon --pidfile=$pidfile $binary $OPTIONS
```

The system should now be configured for Velocity Software products.

Turning off SE Linux

SE Linux is turned off for now by setting the `SELINUX` variable in the file `/etc/selinux/config` from `enforcing` to `permissive`. If this service is a requirement of your security policy, it can be turned on later.

To turn off SE Linux, complete the following steps:

1. Change directory to `/etc/selinux/` and create a backup copy of the `config` file, as shown in the following example:

```
# cd /etc/selinux
# cp config config.orig
```

2. Edit the config file and set `SELINUX` to `permissive`, as shown in the following example:

```
# vi config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

SE Linux is not enforced after the next reboot.

Customizing rc.local

The file `/etc/rc.d/rc.local` is a boot script that is put in place for local configurations. To customize `rc.local`, perform the following steps:

1. Create a backup copy, then edit the file `/etc/rc.d/rc.local`, which is run at boot time, as shown in the following example:

```
# cd /etc/rc.d
# cp rc.local /tmp/rc.local.orig
```

2. Edit the file and add the following lines:

```
# vi rc.local
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
```

```
# want to do the full Sys V style init stuff.
touch /var/lock/subsys/local

# Set system to logoff on halt and power off
chshut halt vmcmd logoff
chshut poff vmcmd logoff

# detach CMS disks to allow for possible live guest relocation
modprobe vmcp
vmcp det 190
vmcp det 191
vmcp det 19d
vmcp det 19e

# Add the sg module for SCSI support of LUNs
modprobe sg
```

Be aware of the following points concerning the additional lines:

- The first two lines set the system to log off the z/VM virtual machine at shut down and power off time by using the **chshut** command. This is useful with the z/VM **SIGNAL SHUTDOWN** feature, in that all Linux systems shut down and log off cleanly when z/VM is shut down.
- The next five lines install the `vmcp` module so CP commands can be issued from Linux then use the module to detach four CMS disks. This is so it is possible to use Live Guest Relocation (LGR) with z/VM 6.2 Single System Image (SSI) because virtual machines with links to CMS disks cannot be relocated.
- The last line adds the `sg` module for SCSI support of LUNs.

Turning on a boot script

There are many ways to clone Linux. One approach is to copy the golden image's disks to those of a new virtual machine, IPL (boot) the new Linux, and have the new Linux configure itself before networking is started. This is the model that is used in this chapter with the file `boot.onetime`. For more information about the source code listing for this script, see Appendix D, "Additional material" on page 357. It reads the same CMS configuration file that the RHEL installer uses: *userID*.CONF on the CMS 191 disk.

To turn on the first boot script, complete the following steps:

1. Set the `boot.onetime` script to be executable by using the **chmod +x** command, as shown in the following example:

   ```
   # cd /etc/init.d
   # chmod +x boot.onetime
   ```

2. Set the `boot.onetime` script to start at boot time by using the **chkconfig** command, as shown in the following example:

   ```
   # chkconfig boot.onetime on
   ```

3. View the link in the following run level 3 startup sequence:

   ```
   # ls -l /etc/rc.d/rc3.d/S*boot.*
   lrwxrwxrwx. 1 root root 22 Nov  6 15:00 /etc/rc.d/rc3.d/S01boot.onetime ->
   ../init.d/boot.onetime
   ```

   The `boot.onetime` script is set to boot early in run level 3 by noting the `S01` prefix.

The golden image is now configured to run a new boot script.

Shutting down the RHEL 6.2 golden image

It is not a good practice to clone a running system, so all golden images are normally shut down. Shut down the golden image now by using the following command:

```
# shutdown -h now
...
```

The RHEL 6.2 golden image should now be ready to be cloned from.

Important: At the time of this writing, an issue was found with RHEL 6 where the system does not shut down. Instead, the shutdown process froze. A Red Hat bugzilla bug, 872702, was opened and is available at this website:

https://bugzilla.redhat.com/show_bug.cgi?id=872702

10.2.4 Copying a REXX EXEC on z/VM for cloning support

The first half of the cloning process that is described in the previous section is the copying of the disks. This can be done from z/VM or from Linux. The example in this book is a REXX EXEC so it is accomplished from z/VM. The source code listing for this EXEC is in Appendix D, "Additional material" on page 357.

It is recommended that you copy the `CLONE EXEC` to the `MAINT 191` disk. You can choose another virtual machine; however, it must have sufficient privileges. The following example shows the FTP process from a DOS session:

```
C:>\ ftp 9.12.4.200
ftp> user MAINT
ftp> put CLONE.EXEC
ftp> quit
```

You should now be ready to try a clone.

10.2.5 Testing the cloning a virtual machine

In section 10.1.2, "Defining virtual machines" on page 203, three virtual machines were created with identically sized and addressed disks as the RHEL 6.2 golden image. In this example, we test cloning from `RH62GOLD` to one of the other virtual machines, `LNXSA2` , by completing the following steps:

1. Log on to `LNXMAINT`.
2. Copy the configuration and parameter files from those of the golden image to those with file names of the target user ID, as shown in the following example:

   ```
   ==> copy rh62gold conf-rh6 d lnxsa2 = =
   ==> copy rh62gold parm-rh6 d lnxsa2 = =
   ```

3. Edit the parameter file and modify the `CMSCONFFILE` variable to point to the new configuration file, as shown in the following example:

   ```
   ==> x lnxsa2 parm-rh6
   root=/dev/ram0 ro ip=off ramdisk_size=40000
   CMSDASD=191 CMSCONFFILE=LNXSA2.CONF-RH6
   vnc vncpassword=12345678
   ```

4. Edit the configuration file and modify the HOSTNAME and IPADDR variables for the new host name and primary IP address. Also, modify the IPADDR2 variable, which the Red Hat installer does not reference but the boot.onetime script does, to that of the secondary IP address, as shown in the following example:

```
==> x lnxsa2 conf-rh6
DASD=100-101,200-202,210-213,300-302
HOSTNAME=lnxsa2.itso.ibm.com
NETTYPE=qeth
IPADDR=9.12.7.4
SUBCHANNELS=0.0.0600,0.0.0601,0.0.0602
NETMASK=255.255.240.0
SEARCHDNS=itso.ibm.com
GATEWAY=9.12.4.1
DNS=9.12.6.7
MTU=1500
PORTNAME=DONTCARE
PORTNO=0
LAYER2=1
IPADDR2=10.1.1.4
```

5. Log on to MAINT.
6. Start the CLONE EXEC with the source user ID and the target user ID as parameters, as shown in the following example:

```
==> clone rh62gold lnxsa2
HCPCQU045E RH62GOLD not logged on
Are you sure you want to overwrite disks on lnxsa2 (y/n)?
y
Trying FLASHCOPY of 1100 to 2100 ...
Command complete: FLASHCOPY 1100 0 10015 TO 2100 0 10015
DASD 1100 DETACHED
DASD 2100 DETACHED
Trying FLASHCOPY of 1101 to 2101 ...
Command complete: FLASHCOPY 1101 0 30049 TO 2101 0 30049
DASD 1101 DETACHED
DASD 2101 DETACHED
Trying FLASHCOPY of 1302 to 2302 ...
Command complete: FLASHCOPY 1302 0 10015 TO 2302 0 10015
DASD 1302 DETACHED
DASD 2302 DETACHED
Starting new clone LNXSA2
Command accepted
```

In this example, FLASHCOPY succeeded so the cloning process took only a few seconds.

7. Quickly log out of MAINT and log on to the new clone. You should see boot messages scrolling on the console. If Linux does not boot, there was a problem copying disks that must be remedied. You should see some from S01boot.onetime similar to those shown in the following example:

```
AUTO LOGON  ***       LNXSA2   USERS = 65
HCPCLS6056I XAUTOLOG information for LNXSA2: The IPL command is verified by the
IPL command processor.
...
dasd-eckd 0.0.0191: DASD with 4 KB/block, 360000 KB total size, 48 KB/track,
linux disk layout
 dasdf:CMS1/  LXM192: dasdf1
```

```
S01boot.onetime: source host name  = rh62gold
S01boot.onetime: source IP address = 9\.12\.7\.2
S01boot.onetime: source IP address 2 = 10\.1\.1\.2
S01boot.onetime: targetParm = LNXSA2.CONF-RH6
S01boot.onetime: target host name = lnxsa2
S01boot.onetime: target IP address = 9.12.7.4
S01boot.onetime: target IP address 2 = 10.1.1.4
S01boot.onetime: Modifying network values
stop on panic configured.
...
```

The newly cloned Linux system should come up with its new host name and IP address. Additionally, the second IP address, set in the **IPADDR2** variable, is used to set up the private interconnect. If the newly cloned Linux does not come up with the new IP address, try the IP address of the golden image. If you do get to that IP address, verify the virtual machine that Linux system is running on by using the **vmcp q userid** command. If it is running on the newly cloned virtual machine, this shows that the disk cloning worked, but the setting of the IP address did not work. This issue must be remedied before you proceed.

If you can log in with SSH to the new IP address, run the **ifconfig** command. You should also see the second IP address on the `eth1` interface, as shown in the following example:

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:00:00:00:01:69
          inet addr:9.12.7.4  Bcast:9.12.15.255  Mask:255.255.240.0
          inet6 addr: fe80::ff:fe00:169/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:459 errors:0 dropped:0 overruns:0 frame:0
          TX packets:80 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:40003 (39.0 KiB)  TX bytes:12740 (12.4 KiB)

eth1      Link encap:Ethernet  HWaddr 02:00:00:00:01:6A
          inet addr:10.1.1.4  Bcast:10.1.1.255  Mask:255.255.255.0
          inet6 addr: fe80::ff:fe00:16a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:202 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5656 (5.5 KiB)  TX bytes:830 (830.0 b)
...
```

The next step is SaaS.

10.3 SaaS for Oracle stand-alone

In section 10.2, “PaaS” on page 205, you configured a system and cloned an RHEL 6.2 Linux to it by using the `boot.firstone` script to modify networking values of the new clone. Similarly, to affect SaaS, the `boot.oracle` script was also copied. It is set to run late in run level 3 because many other services, such as, the network, are required to be running first.

Up to this point, only traditional ECDK minidisks were used for disk storage. This is a common practice for Linux on System z as minidisks are easy to virtualize. It is also common to use SCSI/FCP disks for Oracle data because the size requirements can be large. SCSI/FCP disk are not as easy to virtualize because of the need to identify the world wide port number (WWPN) and the Logical Unit Number (LUN). As the `IPADDR2` variable was added to the RHEL 6 configuration file, additional variables also can be added for identifying target SCSI FCP disks. In this example, two disks are added to the system and a logical volume is created for Oracle data. In addition, there are two paths over two CHPIDs to each disk. This is so that multipathing can be set up so to eliminate a single point of failure. This environment is shown in Figure 10-3.

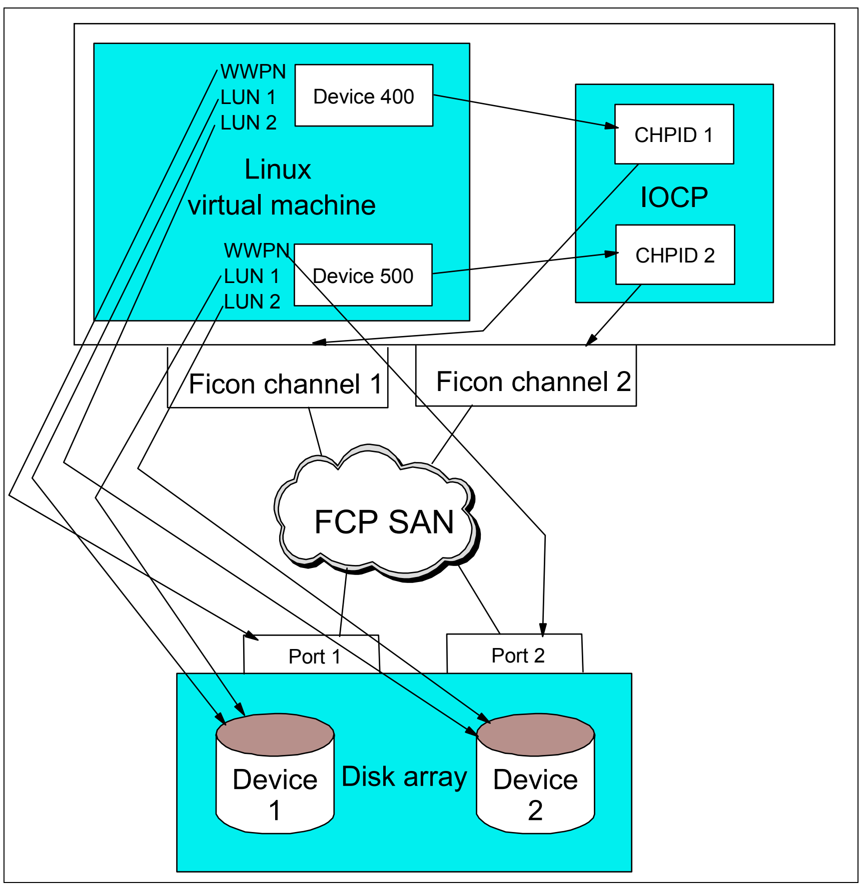


Figure 10-3 FCP/SCSI disks’ WWPNs and LUNs

At a high level, the SaaS’ first boot script, `boot.oracle`, performs the following tasks:

- Defines users and groups for Oracle
- Installs all co-requisite RPMs
- Configures the Network Time Protocol (NTP)
- Sets limits for the system and then the `oracle` and `grid` users
- Sets kernel parameters per Oracle requirements

- Configures FCP disks for Oracle data storage
- Creates a logical volume from the two FCP disks (for Oracle SI)
- Mounts the logical volume over a new directory (for Oracle SI)

To set up SaaS for Oracle SI by using the methodology described, complete the tasks that are described in this section.

10.3.1 Configuring a Linux system for the Oracle boot script

This section describes the second boot script that reads the same CMS configuration file that the RHEL installer uses: `userID.CONF` on the Linux system's read-only `191` disk, which is the `LNXMAINT 192` disk.

To configure the golden image for the Oracle boot script, complete the following steps:

1. Log on to `LNXMAINT`.
2. Add the following variables to the configuration file for the `LNXSA2` virtual machine:
   - The `FCP400WWPN` and `FCP500WWPN` variables set the WWPNs for each channel path (devices `400` and `500`).
   - The `FCPLUN1` and `FCPLUN2` variables set the LUNs for each of two disks.
   - The `SOFTWARE` variable is set to prepare for Oracle Stand-alone (SI). One of the following values is used:
     - `OracleStandalone`: For Single instance servers
     - `OracleGrid`: For servers that are part of a cluster
   - The `PASSWORD` variable is used later to set Oracle passwords.

   The following values are used in the example environment in this book:

   ```
   DASD=100-101,200-202,210-213,300-302
   HOSTNAME=lnxsa2.itso.ibm.com
   NETTYPE=qeth
   IPADDR=9.12.7.4
   SUBCHANNELS=0.0.0600,0.0.0601,0.0.0602
   NETMASK=255.255.240.0
   SEARCHDNS=itso.ibm.com
   GATEWAY=9.12.4.1
   DNS=9.12.6.7
   MTU=1500
   PORTNAME=DONTCARE
   PORTNO=0
   LAYER2=1
   IPADDR2=10.1.1.4
   FCP400WWPN=0x500507630500c74c
   FCP500WWPN=0x500507630508c74c
   FCPLUN1=0x4010401200000000
   FCPLUN2=0x4011401200000000
   SOFTWARE=OracleStandalone
   PASSWORD=Oracle_01
   ```

3. Confirm that the `boot.oracle` script was copied to the golden image into the `/etc/init.d/` directory. If it was not, do so now.
4. Set the script to be executable by using the **chmod +x** command, as shown in the following example:

   ```
   # chmod +x boot.oracle
   ```

5. Set the script to start at boot time by using the **chkconfig** command, as shown in the following example:

   ```
   # chkconfig boot.oracle on
   ```

6. Shut down the golden image.

After the golden image is cloned and the clone is booted, the `boot.oracle` script runs near the end of run level 3 and reads the configuration file on the `LNXMAINT 192` disk.

10.3.2 Cloning a virtual server

You should be ready to clone a server that self-customizes for an Oracle stand-alone or grid installation.

The `CLONE EXEC` that is provided with the code that is associated with this book assumes the minidisk layout of virtual machines that are defined in this chapter. Therefore, it copies minidisks `100`, `101`, and `302`. If you modified this layout, the `EXEC` must be modified accordingly.

Complete the following steps:

1. Start the `CLONE EXEC` with the golden image user ID as the source and the target user ID, as shown in the following example:

   ```
   ==> clone rh62gold lnxsa2
   HCPCQU045E RH62GOLD not logged on
   Are you sure you want to overwrite disks on lnxsa2 (y/n)?
   y
   Trying FLASHCOPY of 1100 to 2100 ...
   Command complete: FLASHCOPY 1100 0 10015 TO 2100 0 10015
   DASD 1100 DETACHED
   DASD 2100 DETACHED
   Trying FLASHCOPY of 1101 to 2101 ...
   Command complete: FLASHCOPY 1101 0 30049 TO 2101 0 30049
   DASD 1101 DETACHED
   DASD 2101 DETACHED
   Trying FLASHCOPY of 1302 to 2302 ...
   Command complete: FLASHCOPY 1302 0 10015 TO 2302 0 10015
   DASD 1302 DETACHED
   DASD 2302 DETACHED
   Starting new clone LNXSA2
   Command accepted
   Ready; T=0.01/0.02 17:20:06
   AUTO LOGON  ***       LNXSA2   USERS = 65
   HCPCLS6056I XAUTOLOG information for LNXSA2: The IPL command is verified by the
   IPL command processor.
   ```

Attention: If your disk array does not support `FLASHCOPY`, or if the service is not ready (for example, if you ran two clones in a short period), the code falls back to the z/VM **`DDR`** command. The output is different and the process is much slower, as shown in the following example:

```
==> clonesa2
HCPCQU045E RH62GOLD not logged on
Are you sure you want to overwrite disks on LNXSA2 (y/n)?
y
Trying FLASHCOPY of 1100 to 2100 ...
Command complete: FLASHCOPY 1100 0 10015 TO 2100 0 10015
DASD 1100 DETACHED
DASD 2100 DETACHED
Trying FLASHCOPY of 1101 to 2101 ...
HCPCMM296E Status is not as required - 1101; an unexpected condition
HCPCMM296E occurred while executing a FLASHCOPY command, code = AE.
FLASHCOPY failed, falling back to DDR ...
z/VM DASD DUMP/RESTORE PROGRAM
HCPDDR696I VOLID READ IS 0X0101
HCPDDR696I VOLID READ IS 0X0101
COPYING   0X0101
COPYING DATA  10/21/12 AT 12.15.11  GMT FROM 0X0101 TO 0X0101
INPUT CYLINDER EXTENTS      OUTPUT CYLINDER EXTENTS
      START       STOP         START        STOP
...
```

2. Log off `LNXMAINT`.
3. Log on to the target virtual machine. You should see that the network is configured again, as shown in the following example:

```
...
S01boot.onetime: this userID = LNXSA2
...
```

Later in the run level, you should see the following output from `boot.oracle`:

```
...
S98boot.oracle: Searching for SOFTWARE variable in parameter file LNXSA2.CONF-RH6
S98boot.oracle: SOFTWARE variable = OracleStandalone
S98boot.oracle: Preparing for Oracle standalone installation
S98boot.oracle: Creating Oracle groups and users
S98boot.oracle: cmd: groupadd -g 198 oinstall
S98boot.oracle: cmd: groupadd -g 199 asmadmin
S98boot.oracle: cmd: groupadd -g 201 dba
S98boot.oracle: cmd: groupadd -g 205 asmdba
S98boot.oracle: cmd: groupadd -g 207 asmoper
...
S98boot.oracle: cmd: chown -R oracle.oinstall /opt/oracle
S98boot.oracle: *
S98boot.oracle: ********************************************************
S98boot.oracle: Sucessfully completed!
S98boot.oracle: ********************************************************
dasd-eckd 0.0.0191: No operational channel path is left for the device
...
```

4. Start an SSH session to the newly cloned Linux system.

5. Change the directory to /tmp/ and review the output file, /tmp/boot.oracle.out.

The system should now be prepared for an Oracle database installation.

10.3.3 Silently installing Oracle Database

Oracle can be installed manually through a graphical interface, or silently with the use of a response file. A silent installation is probably less labor-intensive and more reliable.

In this example, a Linux system was cloned to the virtual machine named LNXSA2. The configuration file includes the variable SOFTWARE=OracleStandalone. When the boot script boot.oracle read this, it prepares the system for an Oracle SI installation.

To perform a silent install of Oracle Database, complete the tasks that are described in the following sections.

Preparing the response file

A database response file named database.rsp is included with the files that are associated with this book. Complete the following steps to view this file:

1. Change to the /tmp/ directory and make oracle the owner of the file and oinstall the group, as shown in the following example:

```
# cd /tmp
```

2. View the contents of the file by using the **cat** command, as shown in the following example:

```
# cat database.rsp
oracle.all.responseFileVersion=/oracle/all/rspfmt_dball_response_schema_v11_2_0
oracle.all.option=ALL_DB_AND_CONFIG
ORACLE_HOSTNAME=xxxx
UNIX_GROUP_NAME=oinstall
INVENTORY_LOCATION=/opt/oraInventory
SELECTED_LANGUAGES=en
ORACLE_HOME=/opt/oracle/11.2
ORACLE_BASE=/opt/oracle
oracle.all.db.installedition=EE
oracle.all.db.EEOptionsSelection=false
oracle.all.db.optionalComponents=oracle.rdbms.partitioning:11.2.0.3.0,oracle.or
aolap:11.2.0.3.0,oracle.rdbms.dm:11.2.0.3.0,oracle.rdbms.dv:11.2.0.3.0,oracle.r
dbms.lbac:11.2.0.3.0,oracle.rdbms.rat:11.2.0.3.0
oracle.all.db.DBA_GROUP=dba
oracle.all.db.OPER_GROUP=dba
oracle.all.db.isRACOneall=false
oracle.all.db.config.starterdb.type=GENERAL_PURPOSE
oracle.all.db.config.starterdb.globalDBName=rac.us.oracle.com
oracle.all.db.config.starterdb.SID=rac
oracle.all.db.config.starterdb.characterSet=AL32UTF8
oracle.all.db.config.starterdb.memoryOption=true
oracle.all.db.config.starterdb.memoryLimit=2000
oracle.all.db.config.starterdb.allExampleSchemas=true
oracle.all.db.config.starterdb.enableSecuritySettings=false
oracle.all.db.config.starterdb.password.ALL=xxxx
oracle.all.db.config.starterdb.control=DB_CONTROL
oracle.all.db.config.starterdb.automatedBackup.enable=false
oracle.all.db.config.starterdb.storageType=FILE_SYSTEM_STORAGE
```

```
oracle.all.db.config.starterdb.fileSystemStorage.dataLocation=/oradata
oracle.all.db.config.asm.ASMSNMPPassword=xxxx
SECURITY_UPDATES_VIA_MYORACLESUPPORT=false
DECLINE_SECURITY_UPDATES=true
oracle.installer.autoupdates.option=SKIP_UPDATES
```

3. Use the **grep** command to view variables that must be set for your enterprise, as shown in the following example:

```
# grep xxxx database.rsp
ORACLE_HOSTNAME=xxxx
oracle.all.db.config.starterdb.password.ALL=xxxx
oracle.all.db.config.asm.ASMSNMPPassword=xxxx
```

   These are replaced by the `boot.onetime` and `boot.oracle` first boot scripts based on variables set in the CMS `CONF` file. The other values should be correct for the settings that are made by the `boot.oracle` script. The values for `ORACLE_HOME` and `ORACLE_BASE` point to `/opt/` over which a large logical volume mounted.

Running the silent installer

To run the silent installer, complete the following steps:

1. Mount the NFS directory with the Oracle binaries. In this example, it is on the NFS server `9.12.5.131` in the `/zCode/` directory. It is mounted over the local directory `/mnt/`, as shown in the following example:

```
# mount 9.12.5.131:/zCode /mnt -o vers=4
```

2. Switch the user to Oracle by using the **su -** command, which runs the user's profile, as shown in the following example:

```
# su - oracle
```

3. Verify that the size of the target directory is sufficient, as shown in the following example:

```
$ df -h | grep opt
/dev/mapper/opt_vg-opt_lv
                       21G  172M   20G   1% /opt
```

   In this example, there is 20 GB of free space.

4. Change to the directory with the binaries for Oracle database. In the following example, it is `database/`:

```
# cd /mnt/database
```

5. Run the **runInstaller** command with the following parameters:

```
$ ./runInstaller -silent -force -ignorePrereq -responseFile ~/database.rsp
Starting Oracle Universal installer...
Checking Temp space: must be greater than 80 MB.   Actual 923 MB    Passed
Checking swap space: must be greater than 150 MB.   Actual 7803 MB    Passed
Preparing to launch Oracle Universal installer from
/tmp/Oraall2012-11-10_06-07-16AM. Please wait ...$ You can find the log of this
all session at:
 /opt/oraInventory/logs/installactions2012-11-10_06-07-16AM.log
```

6. This process runs for some time. It is possible to use the **tail --follow** command to monitor the log file as messages are added. Eventually, you should see a success message, as shown in the following example:

```
The installation of Oracle Database 11g was successful.
Please check '/opt/oraInventory/logs/silentall2012-11-10_06-07-16AM.log' for
more details.

As a root user, execute the following script(s):
        1. /opt/oraInventory/oraRoot.sh
        2. /opt/oracle/product/11.2.0/dbhome_1/root.sh

Successfully Setup Software.
```

7. Exit from the `oracle` user to return to the `root` shell, as shown in the following example:

```
$ exit
```

8. Run the first script, as shown in the following example:

```
# /opt/oraInventory/oraRoot.sh
Changing permissions of /opt/oraInventory.
Adding read,write permissions for group.
Removing read,write,execute permissions for world.

Changing groupname of /opt/oraInventory to oinstall.
The execution of the script is complete.
```

9. Run the second script, as shown in the following example:

```
# /opt/oracle/product/11.2.0/dbhome_1/root.sh
Check
/opt/oracle/product/11.2.0/dbhome_1/all/root_lnxsa2.itso.ibm.com_2012-11-10_06-
52-22.log for the output of root script
```

10. Check the output of the second script, as shown in the following example:

```
# cat
/opt/oracle/product/11.2.0/dbhome_1/all/root_lnxsa2.itso.ibm.com_2012-11-10_06-
52-22.log

Performing root user operation for Oracle 11g

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /opt/oracle/product/11.2.0/dbhome_1

Creating /etc/oratab file...
Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root script.
Now product-specific root actions will be performed.
Finished product-specific root actions.
```

This output shows that the script was successful.

11. Verify that Oracle is working by using the **sqlplus** command, as shown in the following example:

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.3.0 Production on Mon Nov 12 10:54:05 2012

Copyright (c) 1982, 2011, Oracle.  All rights reserved.

Connected to an idle instance.
```

Note the Oracle release number. In this example, it is 11.2.0.3.0.

12. Quit out of the SQL environment, as shown in the following example:

```
SQL> quit
Disconnected
```

This shows that Oracle is installed and working. You should now be ready to start working with Oracle SI.

11

Provisioning an Oracle environment

Attention: The following command conventions are used in this chapter:

- Linux commands that are running as root are prefixed with #
- Linux commands that are running as non-root are prefixed with $

This chapter describes the steps to provision an Oracle Linux guest by using Tivoli products.

This chapter includes the following topics:

- Introduction
- Customizing a new Linux reference for Oracle
- Optimizing the Linux environment for Oracle workload
- Linux configuration for Tivoli Service Automation Manager environment
- Installing a new Oracle Single Instance database and recording the Silent Install file on a test server
- Customizing a script
- Creating the Tivoli Service Automation Manager and Tivoli Provisioning Manager objects and workflows for PaaS provisioning
- Summary

11.1 Introduction

Many IBM System z customers use Linux on System z in a z/VM hypervisor environment to achieve the benefits of server virtualization. While some of these customers used home-grown tools to install, configure, and manage their Linux servers, there is a need for standardized management tools to automate these system management tasks.
IBM Tivoli Service Automation Manager provides such a set of tools. This solution enables customers to rapidly create, configure, provision, and de-provision Linux on System z servers running in the IBM z/VM host environment. It also provides the tools to help you make a staged entrance into *cloud computing*. Cloud computing is an IT model that facilitates better use of existing IT resources with fewer provisioning processes, lower capital, and operating expenses.

Tivoli Service Automation Manager helps the automated provisioning, management, and de-provisioning of cloud resources, consists of hardware servers, networks, operating systems, middleware, and application-level software. Several virtualization environments (hypervisors) are supported while individual virtual server provisioning.

Tivoli Service Automation Manager helps you define and automate services that are lifecycle-oriented; for example, a service to establish and administer an IT server network for a limited period to satisfy increased demand for processing capacity, or to serve as a test environment. Predefined service definitions determine the overall framework for the services. The actual service instances are requested by using these service definitions.

The Self-Service Virtual Server Management environment is used by the cloud users to request provisioning and manage the virtual environments. Tivoli Service Automation Manager provides a Self-Service Virtual Server Management environment, which allows a user to request the provisioning of projects that consists of virtual servers that are based on IBM System x, System p®, or System z and the WebSphere CloudBurst® Appliance product.

Figure 11-1 on page 233 shows the following components of Tivoli Service Automation Manager:

- The *Managed From* component model refers to resources that are needed for Tivoli Services Automation Manager (Tivoli Service Automation Manager/Tivoli Provisioning Manager).
- The *Managed Through* component model refers to z/VM, MAPSRV, Linux Master, DIRMAINT, and VSMSERVE.
- The *Managed To* component model refers to the Linux instances to be provisioned in z/VM.

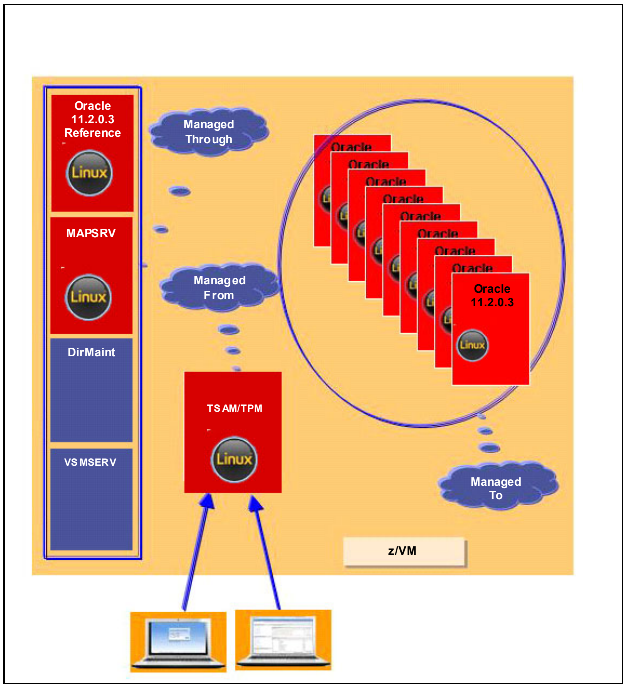


Figure 11-1 ITSO Tivoli Architecture on Linux System

> **Note:** This chapter is not meant to replace the IBM z/VM, Linux, Tivoli Service Automation Manager, or Oracle Databases Enterprise Server products documentation. z/VM, Linux, or Tivoli Service Automation Manager installations are not in the scope of this chapter.

For more information about how to deploy a Tivoli provisioning environment on System z, see *Tivoli Service Automation Manager Version 7.2.2 - Installation and Administration Guide,* SC34-2657.

In this chapter, we focus on Oracle for Linux on System z provisioning automation environment. The following methods can be used to provision an Oracle Linux guest by using Tivoli Service Automation Manager:

- Define a new offering in the Tivoli Service Automation Manager Service's catalog and expose it through the Web User Interface.
- Use Tivoli Service Automation Manager's predefined service Project Linux Servers under z/VM and use an Oracle Image for Oracle middleware installation. (We used this deployment option in this chapter.)

The following steps are developed later in this chapter:

- Create a SUSE Linux Enterprise Server 11 SP1 guest.
- Customize the guest for Oracle and Tivoli Service Automation Manager.
- Install a new Oracle Single Instance Database and record the Silent Install file on a test server.
- Customize a shell script (`preporacle.sh`) to check Linux environment parameters and run the Oracle silent installation.

- Create the Tivoli Service Automation Manager/Tivoli Provisioning Manager objects and Workflows.
- Provision a new Oracle 11gR2 Linux PaaS (Platform as a Service), ready for DB installation.

11.2 Customizing a new Linux reference for Oracle

This section provides information about how to set up a Linux on z/VM environment for Oracle workload (based on general recommendation for Linux on System z) so that Tivoli Service Automation Manager can provision and manage virtual servers with the z/VM hypervisor.

11.2.1 Requirements

The minimum requirement to install Oracle 11gR2 is SUSE Linux Enterprise Server 10 SP3 (or later). Kernel -2.6.16.60-0.54.5 or later is required for an Oracle 11gR2 installation. SUSE Linux Enterprise Server 11 SP1 (2.6.32.12-0.7) + is available and is preferable for Oracle 11gR2 because it incorporates various features of System z hardware.

Verify the release by using the **`cat /proc/version`** command.

The initial 11.2.0.2 base release software can be downloaded from E-Delivery and is available at this website:

http://www.oracle.com/technetwork/database/enterprise-edition/downloads/112020-zlinux64-352074.html

11.2.0.3 Patchset[1] can be downloaded from My Oracle Support (MOS), Patches and Updates Patch:10404530. Check the readme file to determine which `.zip` files you require.

11.2.2 System environment

In this section, we describe the system environment.

System and operating systems

We used following components of the environment:

- IBM zEnterprise EC12 (zEC12)
- z/VM 6.1
- Linux: SUSE Linux Enterprise Server 11 SP1

The following Oracle Guest definition components were used:

- Four Logical Processors
- 6 GB Virtual Memory
- 25000 Cylinders Minidisk (approximately 18 GB)

One minidisk is defined and contains two partitions: `root` (including Oracle binaries) and `swap`.

We want to deploy Oracle instances that are ready to deploy a database, whatever is the size of the data to deploy. More Linux Logical Volumes or Oracle Automatic Storage Management (ASM) Volumes groups are defined later (which is outside the scope of this chapter).

[1] In line with other platforms, it is now a full release that consists of six `.zip` files

Figure 11-2 shows the Oracle Reference Disk Layout that is based on our installation.

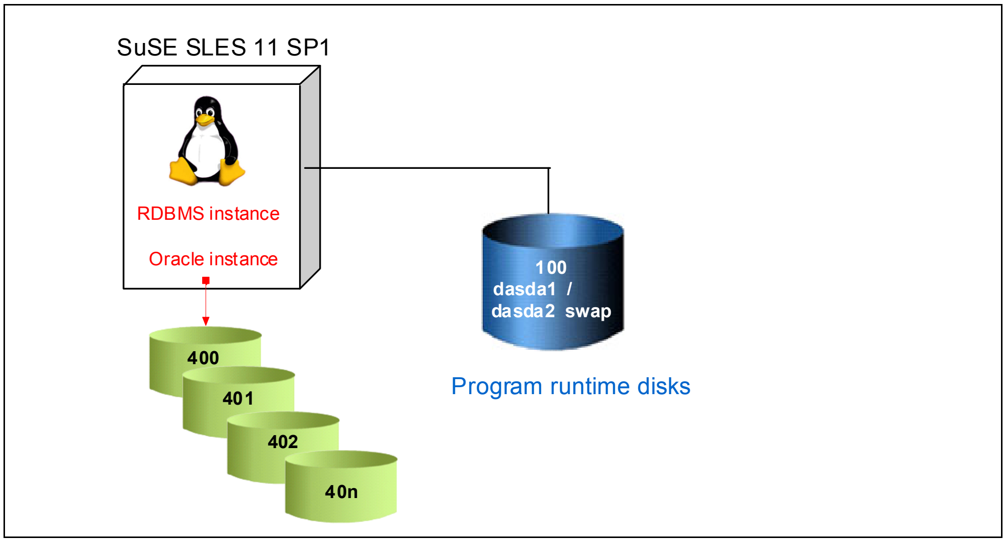


Figure 11-2 Oracle Reference Disk Layout

Software

The following software was used:

- IBM Tivoli Service Automation Manager 7.2.2
- IBM Tivoli Provisioning Manager 7.0.2
- IBM WebSphere Application Server 6.2
- Oracle 11.2.0.3.

The software stack is shown in Figure 11-3.

Version	Service Desk Integration MEA 7.2.0.00 Build 20100419D DB Build V7200-03 TPM PMP 7.2.0.2-IF00001 Build U3415-20110726 DB Build V7221-05 SRM Service Request Management 7.2.0.1 Build 20100316D2 DB Build V7201-01 RDP PMP 7.2.2.0 Build 20110721D DB Build V750-07 IBM Tivoli Change Management Content 7.2.0.00 Build 20091111D DB Build V7200-14 IMBot SRM Service Desk 7.2.0.00 Build 20100419D DB Build V7200-03 SRM Service Desk Content - Best Practices 7.2.0.1 Build 20100316D2 DB Build V7201-01 Service Automation Manager - CCMDB Integration PMP 7.2.2.0 Build 20110721D DB Build V750-03 SRM Service Desk Content - Classification 7.2.0.0 Build 20100419D DB Build V7200-05 Service Automation Manager PMP 7.2.2.0 Build 20110721D DB Build V750-07 IBM Tivoli Common Process Components 7.2.0.01 Build 20100109D DB Build V7201-02 Service Automation Manager Configuration PMP 7.2.2.0 Build 20110721D DB Build V750-06 SRM Service Desk Content - Best Practice Users 7.2.0.0 Build 20100419D DB Build V7200-05 SRM Service Desk 7.2.0.1 Build 20100316D2 DB Build V7201-04 SRM Screen Capturer 7.2.0.0 Build 20100419D DB Build V7200-02 IBM Tivoli Common Process Components for Service Providers 7.2.0.0 Build 20091105D DB Build V7200-04 SRM Problem Management 7.2.0.1 Build 20100316D2 DB Build V7201-01 SRM Solution 7.2.0.1 Build 20100316D2 DB Build V7201-01 SRM Incident Management 7.2.0.1 Build 20100316D2 DB Build V7201-01 SRM SLA Hold 7.2.0.1 Build 20100316D2 DB Build V7201-03 SRM Service Catalog Base 7.2.0.1 Build 20100316D DB Build V7201-05 Advanced Workflow Components 7.3.0.0 Build 201106290658 DB Build V7300-06 SRM Service Desk for Service Providers 7.2.0.0 Build 20100419D DB Build V7120-03 SRM Search 7.2.0.1 Build 20100316D2 DB Build V7201-02 CI Reservation PMP 7.2.2.0 Build 20110721D DB Build V711e-32 SRM Survey Management 7.2.0.1 Build 20100316D2 DB Build V7201-01 IBM Tivoli UI Widgets 7.2.0.01 Build 201002251643 DB Build V7201-02 IBM Tivoli Change Management 7.2.0.01 Build 20100215D2 DB Build V7201-02 ServiceProvider 7.1.1.1 Build BUILD DB Build V7121-03 HFDB Build HF7121-01 SRM Service Catalog Content 7.2.0.1 Build 20100316D DB Build V7201-06 Base Services 7.1.1.8-LA20110712-1206 Build 20110105-1024 DB Build V7118-37 HFDB Build HF7118-15
Server OS	Linux 2.6.16.60-0.54.5-default

Figure 11-3 Software stack for Tivoli on Linux on System z

11.2.3 New reference guest

In this section, we define a new z/VM Linux user. This guest is the reference for provisioned Oracle guests on SUSE Linux Enterprise Server 11 SP1. For more information about installing Linux on System z, see the Linux documentation and IBM Redbooks.

The process includes the following basic steps:

1. Create the Linux guest in z/VM.
2. Install the Linux master (reference) system.

11.3 Optimizing the Linux environment for Oracle workload

In this section, we describe the sets up that are used to perform some of the Oracle configuration for on Linux on System z. However, a script can be developed to run this task and is used to set up the images in a post-provisioning process.

This step is managed by the `preporacle` shell script. For more information about the sample `preporacle.sh` file, see Appendix D, “Additional material” on page 357.

11.3.1 rpm checker

For Linux guests on System z, Oracle 11gR2 rpm checker should be used to verify the system configuration before Oracle software is installed.

Procedure

Download the appropriate rpm checker from My Oracle Support (MOS) Note 1306465.1. The rpm checker verifies whether the required rpms for Oracle Grid and database are installed. This prevents problems with the installation of Oracle. You must log on to the Oracle secure website and select one of the rpm checkers. S11 Grid Infrastructure/Database RPM checker 11.2.0.2 (1.38 KB) (SUSE Linux Enterprise Server 11 Checker) was used for this chapter. For more information, see this website:

https://support.oracle.com/CSP/main/article?cmd=show&type=ATT&id=1086769.1:DB_S11_11202_ZLINUX

Extract the download `.zip` file and then install the extracted rpm to verify your Linux rpm requirements. The rpm checker does not actually install anything. Instead, the checker uses the dependencies of rpm to check your system. Run the RPM checker command as the root user, if possible.

Call the rpm checker by using the following command:

```
rpm -ivh ora-val-rpm-S11-DB-11.2.0.2-1.s390x.rpm
```

Figure 11-4 shows the output of the rpm checker script.

```
lloraref:/home/armelle/rpm # ls -l
total 20
-rwxrwxrwx 1 root root 8526 Dec  1  2011 cvuqdisk-1.0.9-1.rpm
-rw-r--r-- 1 root root 2817 Mar 30  2011 ora-val-rpm-S11-DB-11.2.0.2-1.s390x.rpm
-rw-r--r-- 1 root root 1408 Oct 15 18:02 ora_val_rpm_S11_11202_1_s390x.zip
lloraref:/home/armelle/rpm # rpm -ivh ora-val-rpm-S11-DB-11.2.0.2-1.s390x.rpm
error: Failed dependencies:
        libaio-devel >= 0.3.109-0.1.46 is needed by ora-val-rpm-S11-DB-11.2.0.2-1.s39
        libaio-devel-32bit >= 0.3.109-0.1.46 is needed by ora-val-rpm-S11-DB-11.2.0.2
        sysstat >= 8.1.5-7.9.56 is needed by ora-val-rpm-S11-DB-11.2.0.2-1.s390x
        glibc-devel >= 2.11.1-0.17.4 is needed by ora-val-rpm-S11-DB-11.2.0.2-1.s390x
        linux-kernel-headers >= 2.6.32-1.4.13 is needed by ora-val-rpm-S11-DB-11.2.0.
        gcc >= 4.3-62.198 is needed by ora-val-rpm-S11-DB-11.2.0.2-1.s390x
        gcc43 >= 4.3.4_20091019-0.7.35 is needed by ora-val-rpm-S11-DB-11.2.0.2-1.s39
        glibc-devel-32bit >= 2.11.1-0.17.4 is needed by ora-val-rpm-S11-DB-11.2.0.2-1
        gcc-32bit >= 4.3-62.198 is needed by ora-val-rpm-S11-DB-11.2.0.2-1.s390x
        gcc43-32bit >= 4.3.4_20091019-0.7.35 is needed by ora-val-rpm-S11-DB-11.2.0.2
        libgomp43-32bit >= 4.3.4_20091019-0.7.35 is needed by ora-val-rpm-S11-DB-11.2
        libstdc++43-devel >= 4.3.4_20091019-0.7.35 is needed by ora-val-rpm-S11-DB-11
        gcc-c++ >= 4.3-62.198 is needed by ora-val-rpm-S11-DB-11.2.0.2-1.s390x
        gcc43-c++ >= 4.3.4_20091019-0.7.35 is needed by ora-val-rpm-S11-DB-11.2.0.2-1
        libstdc++43-devel-32bit >= 4.3.4_20091019-0.7.35 is needed by ora-val-rpm-S11
        libstdc++-devel >= 4.3-62.198 is needed by ora-val-rpm-S11-DB-11.2.0.2-1.s390
        libcap1 >= 1.10-6.10 is needed by ora-val-rpm-S11-DB-11.2.0.2-1.s390x
lloraref:/home/armelle/rpm #
```

Figure 11-4 Example of rpm checker output

Install all missing libraries that are shown in the rpm checker output. Check your installed rpm again, as shown in Figure 11-5.

```
lloraref:/home/armelle/rpm # rpm -ivh ora-val-rpm-S11-DB-11.2.0.2-1.s390x.rpm
Preparing...                ########################################### [100%]
   1:ora-val-rpm-S11-DB     ########################################### [100%]
***********************************************************************
* Validation complete - Your sles11 OS has required rpms for Oracle 11.2   *
***********************************************************************
```

Figure 11-5 Linux with required rpms rpm checker output

11.3.2 Optional rpm for Oracle Grid: cvudisk-1.0.9-1 rpm

Although we do not set up a grid infrastructure in this chapter, it is worth it to have this rpm installed in an Oracle Linux reference.

This step is managed by the `preporacle` shell script.

Procedure

For an Oracle Grid install, install the `cvudisk-1.0.9-1` rpm package from the Oracle 11gR2 distribution media.

The Cluster Verification Utility (CVU) requires root privilege to gather information about the SCSI disks during discovery. A small binary uses the `setuid` mechanism to query disk information as root. This process is a read-only process with no adverse effect on the system. To make this secured, this binary is packaged in the `cvuqdisk` rpm and needs root privilege to install on a machine.

When this package is installed on all the nodes, CVU performs discovery and shared storage accessibility checks for SCSI disks. Otherwise, it complains about the missing package `cvuqdisk`. You can disable the SCSI device check feature by setting the `CV_RAW_CHECK_ENABLED` to `FALSE` in `$CV_HOME/cv/admin/cvu_config` file. CVU does not complain about the missing rpm if this variable is set to `FALSE`.

This rpm can be found in the grid installation disk, which can be sent via FTP it to the target server and installed by using the following command:

```
rpm -iv cvuqdisk-1.0.9-1.rpm
```

11.3.3 Stopping the useless Linux services

To keep the reference image as clean as possible in terms of processor usage, some of these services can be turned off.

There are several services that are started in a SUSE Linux Enterprise Server 11 SP1 system. They can be viewed by using the following **chkconfig** command:

```
chkconfig -l | grep 3:on
```

Turn off useless services by using the **chkconfig** command, as shown in the following example:

```
chkconfig fbset off
chkconfig network-remotefs off
chkconfig postfix off
chkconfig splash off
chkconfig splash_early off
chkconfig smartd off
chkconfig alsasound off
chkconfig kbd off
chkconfig xdm off
```

11.3.4 Hosts file

In this step, the `/etc/host` file is customized.

Complete the following steps to customize the file:

1. Comment out any IPV6.
2. Check whether ARP is used by using the following command:

   ```
   # ifconfig -a
   ```

   The following result should not appear:

   ```
   eth0 Link encap:Ethernet HWaddr 02:00:02:00:00:A2
   inet addr:130.35.55.234 Bcast:130.35.55.255 Mask:255.255.252.0
   inet6 addr: fe80::200:200:100:a2/64 Scope:Link
   UP BROADCAST RUNNING NOARP MULTICAST MTU:1492 Metric:1
   RX packets:5749678 errors:0 dropped:0 overruns:0 frame:0
   TX packets:2799431 errors:0 dropped:0 overruns:0 carrier:0
   collisions:0 txqueuelen:1000
   RX bytes:1414260847 (1.3 GiB) TX bytes:2735238017 (2.5 GiB)
   ```

Important: When the two (public and private) network interfaces for Oracle RAC are configured, you must have ARP enabled (that is, NOARP must not be configured). The `root.sh` script fails on the first node if NOARP is configured.

Oracle requires a host name with a fully qualified domain name and a corresponding entry in the `/etc/hosts` file.

11.3.5 Linux Kernel parameters

As the root user, ensure that the required Kernel Parameters are set.

This step is managed by the `preporacle` shell script.

Procedure

Back up the `sysctl.conf` file before any modification is done by using the ***-p*** option to keep the date, as shown in the following example:

```
cp -p /etc/sysctl.conf /etc/sysctl.conf.orig
```

Several parameters should be reviewed concerning the function of the activity (see the comments in this file).

To check the kernel parameters, enter the following commands:

```
cat /proc/sys/fs/file-max
```

or

```
# sysctl -A | grep file-max (file-max parameter for example)
```

To check all, enter the following command:

```
# sysctl -A > collect_sysctl
```

To add or change the kernel parameters, edit the `sysctl.conf` file. Add or change the following values:

```
# Oracle Kernel Specific parameters
#
#fs.file-max = 512 x oracle processes (for example 6815744 for 13312 processes)
fs.file-max = 6815744
# fs.aio-max-nr = 3145728 (use for large concurrent I/O databases
fs.aio-max-nr = 1048576
#kernel.shmall = set to (sum of all SGAs on system) / 4096 or a default of
2097152
kernel.shmall = 2097152
#kernel.shmmax=MAX (1/2 the virtual RAM, largest SGA_MAX_SIZE/SGA_TARGET on
system)
kernel.shmmax = 4218210304
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
```

```
net.core.wmem_max = 1048576
kernel.spin_retry = 2000
#vm.nr_hugepages = 4000 (Use for large SGAs > 10 GB)
```

> **Note:** The minimum value that is required for `shmmax` is 0.5 GB. However, Oracle recommends that you set the value of `shmmax` to 2 GB for optimum performance of the system.

By specifying the values in the `/etc/sysctl.conf` file, they persist when the system restarts. However, on SUSE Linux Enterprise Server systems, enter the following command to ensure that the system reads the `/etc/sysctl.conf` file when it restarts:

```
# /sbin/chkconfig boot.sysctl on
```

Run **`sysctl -p`** for the kernel parameter changes to take effect.

11.3.6 Creating and verifying the required UNIX groups and Oracle user accounts

In this section, we describe the process to update the Oracle user file descriptors. This step is managed by the `preporacle` shell script.

Procedure

When Oracle 11gR2 Database Server single instance is installed, Oracle recommends creating the following groups:

- dba
- oinstall

It is possible to install with one group (for example, only dba). If you are installing only database executable files, often one user account called `oracle` is created.

To verify that the Linux groups and users were created, you can view the group and password files by using the following commands:

```
# cat /etc/passwd | grep oracle
# cat /etc/group | grep oinstall
# cat /etc/group | grep dba
# /usr/sbin/groupadd -g 1100 oinstall
# /usr/sbin/groupadd -g 1101 dba
# /usr/sbin/useradd -u 1100 -g oinstall -G dba -s /bin/ksh -m oracle
# passwd oracle
```

Updates can be verified with the commands that are shown in the previous example.

11.3.7 Pre-creating user directories for product installs

In this section, we show how to create directories that are needed for Oracle. This step is managed by the `preporacle` shell script.

The Oracle I/O architecture that is defined in this chapter for a Single Instance Database follows the Oracle Flexible Architecture (OFA) requirements, as shown in Figure 11-6 on page 241.

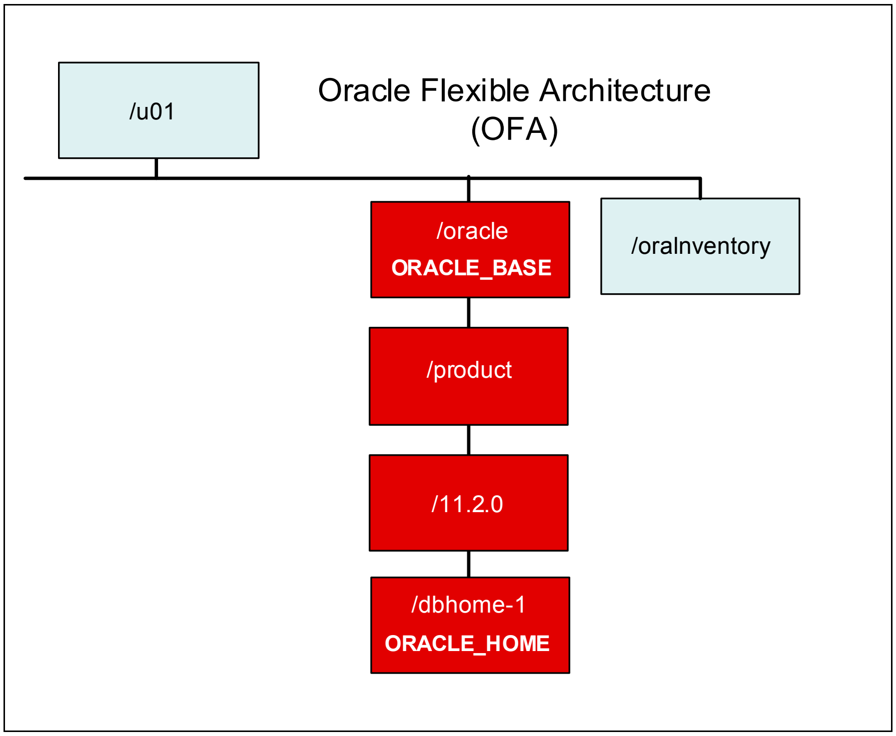


Figure 11-6 Oracle Flexible Architecture

Procedure

Table 11-1 describes the disk space requirements for software files and data files for each installation type on IBM Linux on System z.

Table 11-1 Disk space requirements for software and data files

Installation type	Requirement for software files	Requirement for data files
Enterprise Edition	4.9 GB	2 GB
Standard Edition	4.5 GB	1.5 GB

More disk space on a file system or an Oracle ASM disk group is required for the fast recovery area if you choose to configure automated backups of the following components:

- Oracle executable files

  Oracle software is installed under the `root` file system.

- Oracle data files

  These files can be handled by ASM or be placed in a Linux Logical Volume.

- Oracle data directories that have the correct permission bits

  Create the `ORACLE_BASE` directory that is the Oracle directory tree root, `/u01/oracle` product, as shown in the following example:

  ```
  # mkdir -p /u01/oracle
  # chown -R oracle:oinstall /u01/oracle
  ```

  Create `ORACLE_HOME` directory that handle Oracle software executable files, `/u01/oracle/11.2.0/db_home1/`, as shown in the following example in our case:

  ```
  # mkdir -p /u01/oracle/11.2.0/db_home1/
  ```

```
# chown -R oracle:oinstall /u01/oracle/11.2.0/
# chown -R oracle:oinstall /u01/oracle/11.2.0/db_home1/
```

Create `/u01/oraInventory` directory to handle Oracle referential, as shown in the following example:

```
# mkdir -p /u01/oraInventory
# chown -R oracle:oinstall /u01/oraInventory
```

Create `/u01/oracle/oradata` directory for DB installation, as shown in the following example:

```
# mkdir -p /u01/ oracle/oradata
# chown -R oracle:oinstall /u01/oracle/oradata
```

11.3.8 Setting file descriptors limits for Oracle and grid users

A few open file descriptors (sockets) can significantly reduce the performance and load that Oracle can generate. In this section, we describe the process that is used to update the Oracle user file descriptors.

This step is managed by the preparedly shell script.

Procedure

Complete the following steps to set the file descriptors limits for Oracle and grid users:

1. Edit the `/etc/security/limits.conf` file by adding or modifying the following lines:

   ```
   # oracle
   oracle soft nofile 1024
   oracle hard nofile 65536
   oracle soft nproc 2047
   oracle hard nproc 16384
   oracle soft stack 10240
   oracle hard stack 32768
   # Use memlock for Huge Pages support
   #*      soft    memlock 3145728
   #*      hard    memlock 3145728
   # End oracle
   ```

   To increase the limits at Oracle logon as the `oracle` user, verify the `oracle` user's `.profile` (for example, `/home/oracle/.profile` for KSH users) and ensure that the following lines were added:

   ```
   ulimit -n 65536
   ulimit -u 16384
   ulimit -s 32768
   ```

   Another method is to modify the main system profile by adding these lines to the `/etc/profile` file. Change this if the `oracle` user is using a separate user shell program, such as, csh or bash.

2. When you log in to the Linux machine, the login program reads the `/etc/pam.d/login` file. The following line was included in this file, which instructs the login program to load the `pam_limits.so` module during login.session:

   ```
   session required pam_limits.so
   ```

Browse to the `/etc/pam.d` directory. The `session required pam_limits_.so` setting can be added to `/etc/pam.d/login`, `/etc/pam.d/sshd`, or `/etc/pam.d/su`, depending on whether you want to set limits on login, SSH, or su types of log ins. The PAM module (`pam_limits.so`) is not loaded by default for various applications, such as, login, SSH, or su. By adding it explicitly and loading it, you can limit the login sessions. Ensure that the `/etc/pam.d/sshd`, login, and su file has an entry for `pam_limits.so`.

> **Important:** After any change, test a new login or sshd session before all opened sessions are closed.

11.3.9 Activating the swap in memory (VDisk)

It is recommended that an in-memory swap device is defined. This allows Linux to have a first level of swap on a rapid device. This section shows how to enable several swaps with different levels of priority.

Procedure

Complete the following steps to activate the swap in memory:

1. Ensure that all your Linux production guests define a virtual swap. Check that Linux users are allowed to define `VDISKS` in the `SYSTEM CONFIG` file, as shown in the following example:

   ```
   VDisk, /* Allow VDISKS for Linux swaps */
   Syslim infinite,
   Userlim infinite
   ```

2. Ensure that `SWAPGEN EXEC` is installed in `LNXMAINT`, which is the Linux instance that is dedicated to Linux users' administration.
3. Ensure that `SWAPGEN` is called by the Linux users' `PROFILE EXEC`, at Linux IPL. as shown in the following example:

   ```
   /* PROFILE EXEC for zLinux */
   'CP SET RUN ON'               /* CP READ will not stop server */
   'CP SET PF11 RETRIEVE FORWARD' /* Next command */
   'CP SET PF12 RETRIEVE'        /* Previous command */
   'SWAPGEN 101 250000'      /* VDisk swap space; Q VDSK U*/
   'CP IPL 100 CLEAR'            /* Let's roll */
   ```

4. Enable this virtual device as a swap in Linux (as shown in Figure 11-7) by completing the following steps:
   a. Set the specified device online, as shown in the following example:

      ```
      # chccwdev -e 0.0.101
      ```

   b. Check whether the swap device is available in Linux, as shown in the following example:

      ```
      # lsdasd
      ```

   c. Display the swap summary by device, as shown in the following example:

      ```
      # swapon -s
      ```

```
linux:~ # chccwdev -e 0.0.101
Setting device 0.0.0101 online
Done
linux:~ # lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   7042MB    1802880
0.0.0103   active      dasdb     94:4    ECKD  4096   2109MB    540000
0.0.0102   active      dasdc     94:8    ECKD  4096   7042MB    1802880
0.0.0101   active      dasdd     94:12   FBA   512    244MB     500000
linux:~ # swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasdc1                             partition       7211416 0       -1
```

Figure 11-7 Output of the chccwdev, lsdasd, and swapon commands

5. Edit the `/etc/fstab` file to add the new swap device, which provides a higher priority to the swap device in memory so that the first amounts of swapped data are done to the memory.
6. Set up a Linux swap area on the swap device, as shown in the following example:

   `# **mkswap /dev/disk/by-path/ccw-0.0.0101-part1**`

7. Make available all devices that are marked as swap in `/etc/fstab`, as shown in the following example:

   `# **swapon -a**`

8. Display again the swap summary by device and verify that the new swap was added, as shown in Figure 11-8:

   `# **swapon -s**`

```
linux:~ # mkswap /dev/disk/by-path/ccw-0.0.0101-part1
Setting up swapspace version 1, size = 249992 KiB
no label, UUID=8eb54257-4556-4926-898e-6d6041a0531d
linux:~ # swapon -a
linux:~ # swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasdc1                             partition       7211416 0       -1
/dev/dasdd1                             partition       249988  0       20
```

Figure 11-8 Output of swapon command

11.4 Linux configuration for Tivoli Service Automation Manager environment

To use this new guest as a reference for Tivoli Service Automation Manager deployments, make the changes to the Linux system that are described in this section.

11.4.1 Python package

Make sure that python and python-XML packages are installed.

With root user, run the following command:

`# **rpm -qa | grep -i python**`

11.4.2 Installing rpm for Tivoli Service Automation Manager

Copy `IBM-System-z.MASTER-<version number>.s390x.rpm` from the Tivoli Service Automation Manager management server to the Linux master ID (the Oracle reference) and install it.

Install the rpm by using the following command:

```
# rpm -ivh IBM-System-z.MASTER-<version number>.s390x.rpm
```

In our environment, the rpm name was `IBM-System-z.MASTER-7.2.1-1.s390x`.

11.4.3 Checking mount definitions in /etc/fstab

Check the mount definitions in `/etc/fstab` file, as shown in Figure 11-9. Mounts should happen by label, not by device or ID, which can change in the copied image, as shown in Figure 11-9.

```
l1oraref:/home/armelle/rpm # cat /etc/fstab
/dev/disk/by-path/ccw-0.0.0100-part2 swap                swap       defaults
/dev/disk/by-path/ccw-0.0.0100-part1 /                   ext3       acl,user_xattr
proc                 /proc                proc       defaults              0 0
sysfs                /sys                 sysfs      noauto                0 0
debugfs              /sys/kernel/debug    debugfs    noauto                0 0
devpts               /dev/pts             devpts     mode=0620,gid=5       0 0
```

Figure 11-9 fstab example displaying mounts by label

11.4.4 Disabling the boot menu at IPL

This optional procedure eliminates the 10-second wait for the boot menu during the IPL.

Edit the `:menu` section of `/etc/zipl.conf` and change `prompt=1` to `prompt=0`. Save the change and run the **`.zipl`** command.

11.4.5 Disabling parallel boot option on SUSE Linux Enterprise Server 11

On SUSE Linux Enterprise Server 11, ensure that parallel boot option is disabled by editing the `/etc/sysconfig/boot` file with `RUN_PARALLEL="no"` parameter. Save the file.

11.4.6 Verifying the configuration

Complete the following steps to verify the configuration:

1. Log in to `MAPSRV`.
2. Change to the `/opt/ibm/ztsam/workflow` directory and run the following command with the values corresponding with your environment:

   ```
   ../bin/uhubrpcclient 172.16.0.1 1023 mapauth ntccloud 14091 imagequery L1ORAREF
   ```

This command should return the z/VM directory entry of the Linux reference.

11.5 Installing a new Oracle Single Instance database and recording the Silent Install file on a test server

In this section, we describe how to record the Silent Install file, which is used later for the Oracle installation on the new provisioned instances.

11.5.1 Getting the Patchset for Oracle Database Server 11.2.0.3.0

The Oracle Database version 11gR2 (11.2.0.3) was released in December 2011 as a Patch Set. It is a complete replacement for the 11.2.0.2 version, so you do not need to install 11.2.0.2 for a new installation. You can download the six files on My Oracle Support for Patch 10404530.

11.5.2 Starting the Virtual Network Computing server

An X terminal session is required to install the Oracle software on Linux on System z. *Vncserver* is a wrapper script for Xvnc, the free X server for Virtual Network Computing (VNC). It provides all capabilities of a standard X server, but does not connect to a display. Instead, Xvnc creates a virtual desktop you can view or control remotely by using a VNC viewer.

To start the VNC server, login with `oracle` user and start a `vncserver` session with the following command, as shown in Figure 11-10:

```
$ vncserver
```

This command starts Xvnc on the next available display with the following suitable defaults:

```
$ vncserver :1 invokes Xvnc on display :1
```

Because this is the first time it is started, a password is requested to access your desktops. If needed, the server can be stopped by using the following command:

```
$ vncserver -kill :1
```

where **:1** is the display number, as shown in the start command.

```
l1oradb1:/u01 # vncserver

You will require a password to access your desktops.

Password:
Verify:
Would you like to enter a view-only password (y/n)? n
xauth:  creating new authority file /root/.Xauthority

New 'X' desktop is l1oradb1:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/l1oradb1:1.log

l1oradb1:/u01 # ps -ef | grep vnc
root      2815     1  3 12:11 pts/0    00:00:00 Xvnc :1 -desktop X -httpd /usr/
t 120000 -rfbauth /root/.vnc/passwd -rfbport 5901 -fp /usr/share/fonts/misc:uns
100dpi:unscaled,/usr/share/fonts/Type1,/usr/share/fonts/URW,/usr/share/fonts/Sp
set
root      2840  2384  0 12:11 pts/0    00:00:00 grep vnc
```

Figure 11-10 Output of vncserver command

11.5.3 Starting a VNC client

Start a VNC client session by using appropriate client, free TightVNC[2] viewer in our case, as shown in Figure 11-11.

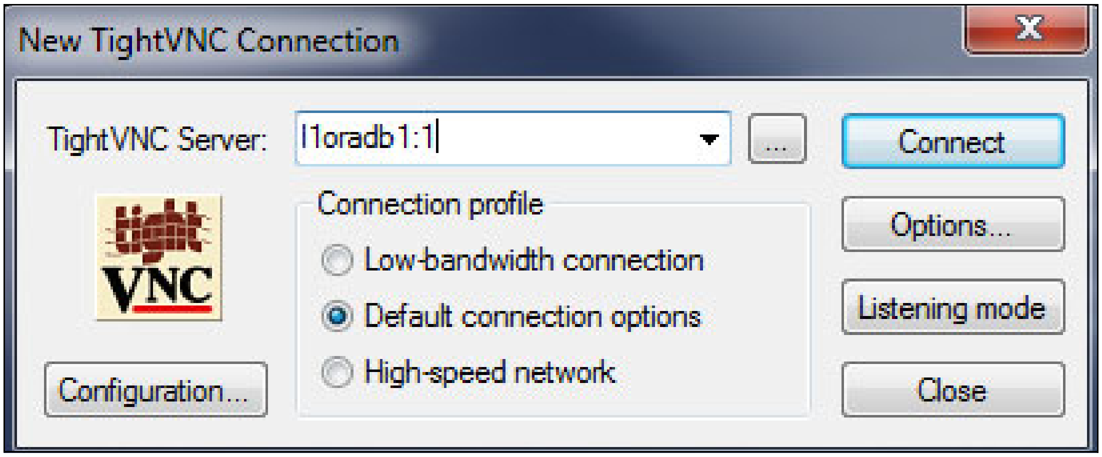


Figure 11-11 New TightVNC Connection panel

To get a better display with your VNC client, choose **Hextile** encoding instead of tight. Go to the VNC Connection Options panel and select **Hextile** from the Format and Encodings drop-down list, as shown in Figure 11-12.

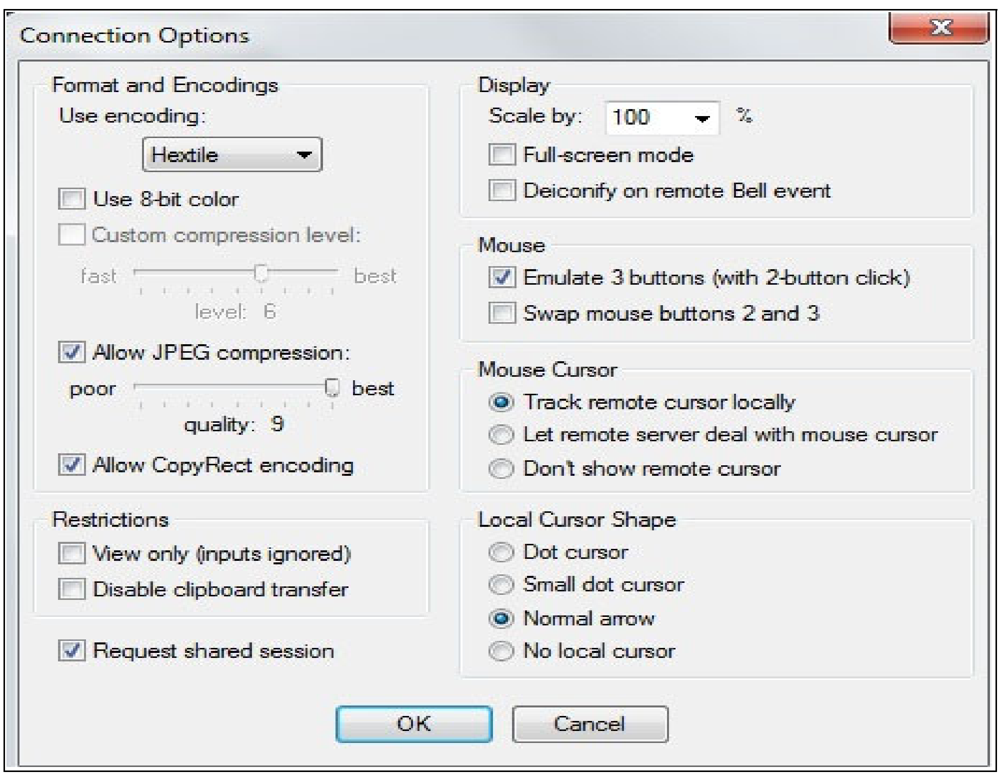


Figure 11-12 Connection Options panel

2 Available at this website: http://www.tightvnc.com

11.5.4 Installing Oracle Database

In this section, we describe how to install the Oracle binaries.

In the Oracle 11.2.0.3.0 patchset, the code is provided in six `.zip` files. Extracting these files on an NFS server creates the directories that are listed in Figure 11-13.

```
lloradb1:/mnt # ls -l
total 4534912
drwxrwxr-x  6 root root       4096 Dec  1  2011 client
drwxrwxr-x  8 root root       4096 Dec  1  2011 database
drwxrwxr-x 19 root root       4096 Dec  1  2011 deinstall
drwxrwxr-x  6 root root       4096 Dec  1  2011 examples
drwxrwxr-x  8 root root       4096 Dec  1  2011 grid
drwx------  2 root root      16384 Sep 14 12:21 lost+found
-rw-r--r--  1 root root 1469666636 Sep 12 10:02 p10404530_112030_Linux-zSer_1of6.zip
-rw-r--r--  1 root root 1116582187 Sep 12 10:46 p10404530_112030_Linux-zSer_2of6.zip
-rw-r--r--  1 root root  790349194 Sep 12 11:00 p10404530_112030_Linux-zSer_3of6.zip
-rw-r--r--  1 root root  638675679 Sep 12 11:15 p10404530_112030_Linux-zSer_4of6.zip
-rw-r--r--  1 root root  490515254 Sep 12 11:11 p10404530_112030_Linux-zSer_5of6.zip
-rw-r--r--  1 root root  133337969 Sep 12 11:01 p10404530_112030_Linux-zSer_6of6.zip
```

Figure 11-13 Listing Oracle installation directory contents

Complete the following steps to install the Oracle database:

1. Log on with `oracle` user.
2. Browse to the `/mnt/database` directory.
3. Run the `runInstaller` script program to start the installation, as shown in the following example:

   `$ ./runInstaller`

   Figure 11-14 on page 249 shows the kind of errors that can be encountered with `runInstaller` script if the `oracle` user is not logged on or if the `DISPLAY` is not set in the Oracle environment. To fix this type of issue, the vncserver can be started by `oracle` user instead of `root`.

```
X Desktop
l1oradb1:/mnt/database # ls
doc       readme.html  rpm            sshsetup  welcome.html
install   response     runInstaller   stage
l1oradb1:/mnt/database # ./runInstaller

The user is root. Oracle Universal Installer cannot continue installation if the
 user is root.
: No such file or directory
l1oradb1:/mnt/database # su - oracle
oracle@l1oradb1:/home/oracle> cd /mnt/database/
oracle@l1oradb1:/mnt/database> ./runInstaller
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 80 MB.   Actual 3554 MB    Passed
Checking swap space: must be greater than 150 MB.   Actual 6386 MB    Passed
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2012-09-21_12
-20-47PM. Please wait ...
DISPLAY not set. Please set the DISPLAY and try again.
Depending on the Unix Shell, you can use one of the following commands as exampl
es to set the DISPLAY environment variable:
- For csh:                      % setenv DISPLAY 192.168.1.128:0.0
- For sh, ksh and bash:         $ DISPLAY=192.168.1.128:0.0; export DISPLAY
Use the following command to see what shell is being used:
        echo $SHELL
Use the following command to view the current DISPLAY environment variable setti
ng:
        echo $DISPLAY
- Make sure that client users are authorized to connect to the X Server.
To enable client users to access the X Server, open an xterm, dtterm or xconsole
 as the user that started the session and type the following command:
% xhost +
To test that the DISPLAY environment variable is set correctly, run a X11 based
program that comes with the native operating system such as 'xclock':
        % <full path to xclock.. see below>
If you are not able to run xclock successfully, please refer to your PC-X Server
 or OS vendor for further assistance.
Typical path for xclock: /usr/X11R6/bin/xclock
oracle@l1oradb1:/mnt/database>
```

Figure 11-14   runInstaller error

If no errors were encountered, the output of the runInstaller script looks similar to what is shown in Figure 11-15.

```
oracle@l1oradb1:/mnt/database> . ./runInstaller
dirname: invalid option -- 'k'
Try `dirname --help' for more information.
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 80 MB.   Actual 3164 MB    Passed
Checking swap space: must be greater than 150 MB.   Actual 6386 MB    Passed
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2012-09-21_12
-44-54PM. Please wait ...oracle@l1oradb1:/mnt/database>
```

Figure 11-15   Running the runInstaller script

4. Complete the following steps:

   a. Start with the Configure Security Updates step as shown in Figure 11-16, which provides the option to download software updates for the installation. Follow the next 10 panels to configure the new Oracle server, and finish with Summary panel, as shown on Figure 11-18 on page 251.

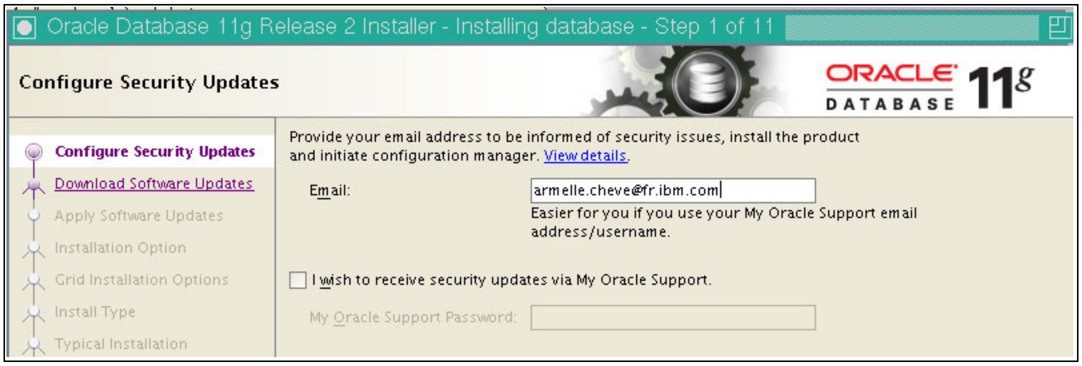


Figure 11-16 First panel of runInstaller

Several components might be missing in the environment to be compliant with Oracle installation. The panel that is shown in Figure 11-17 on page 251 shows a missing rpm. This is a known problem.

You should be aware of the following problems when you are installing upgrades to 11.2.0.3:

– Missing `LIBSTDC++-33-3.2.3-47.3`, which causes the installer to fail.

– Prerequisite check fails in Grid installation concerning missing rpm `compat-libstdc++-33.3.2.3-47.3` (or not installed)

  Select **Ignore All**, click **Next**, and then click **Yes**.

> **Prerequisite checks warning:** Refer to the following documents to fix the missing rpm issues:
>
> - *11.2.0.3 Prereq check warning for missing compatlibstdc++-33.3.2.3-47.3 on IBM: Linux on System z on SLES 11*, Doc ID 1383381.1
> - *Getting Started - 11gR2 Grid Infrastructure, SI (Single Instance), ASM and DB (IBM: Linux on System z)*, Doc ID 1306465.1
> - *Ensure that you have prerequisite RPMs to install Oracle Database and AS10g (midtier) on IBM: Linux on System z (s390x)*, ID 1086769.1

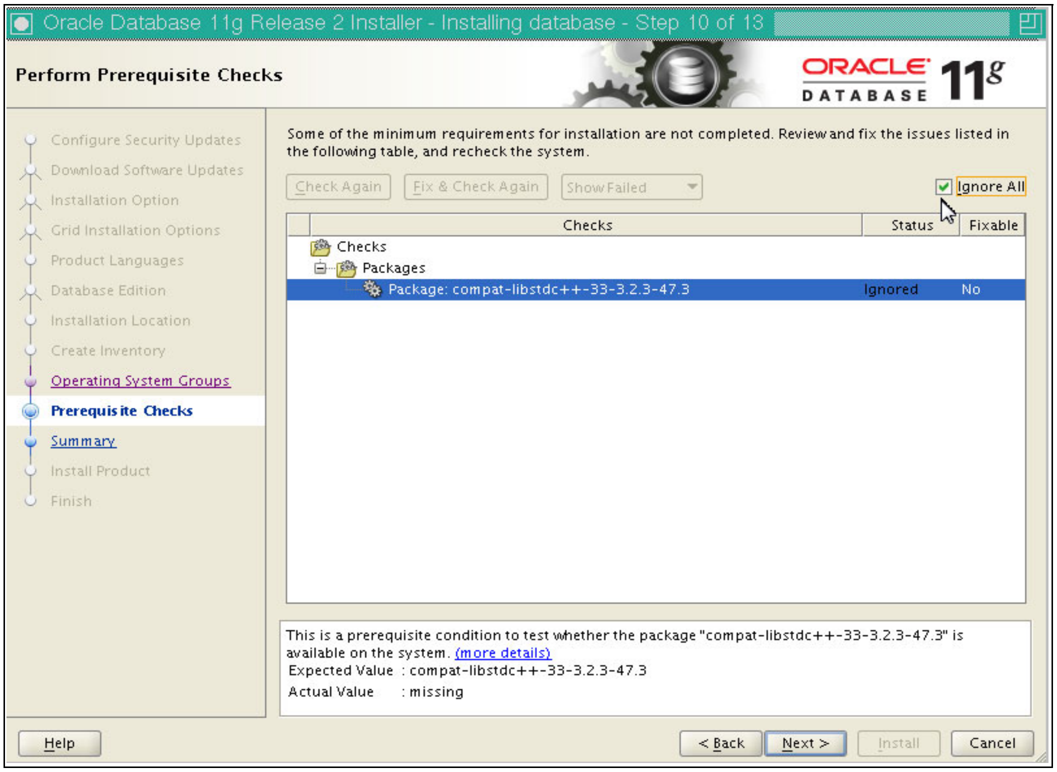


Figure 11-17 Known problem during installation: Missing rpm

5. Click **Save Response File** in the Summary panel, as shown in Figure 11-18.

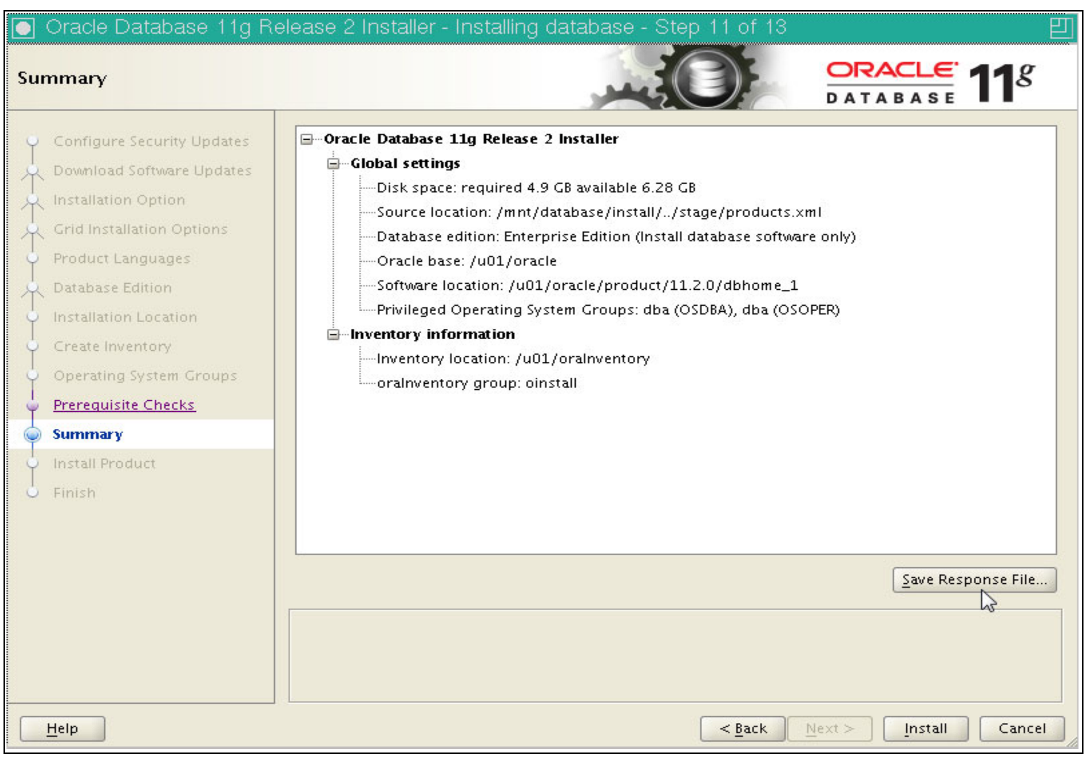


Figure 11-18 Summary panel

6. Accept the home directory as the target directory for the response file for access authorization reasons. Click **Install**.
7. Run the `orainstRoot` shell to change access permissions for the Oracle Referential.
8. Run the root shell script. This script contains all the product installation actions that require root privileges.
9. Return to the Execute Configuration Scripts panel and click **OK**. The Install Product Status for Execute Root Scripts is now green. Click **Next**.

   The Finish panel displays the message, "The installation of Oracle Database was successful." Click **Close**.

   The installation log can be found in the following directory: `/u01/oraInventory/logs/installActionsInstallDate.log`.

11.5.5 Replaying the installation in Silent Mode

Complete the following steps to test the Silent Install response file on the test server:

1. Start the silent mode installation, as shown in the following example:

```
$ ./runInstaller -ignorePrereq -silent -force (-showProgress) -responseFile
/home/oracle/db-install-11.2.0.3-SLES11SP1.rsp
```

> **Note:** This part is included in the `preporacle.sh` script that is started by the Tivoli Service Automation Manager/Tivoli Provisioning Manager workflow. The **`showProgress`** option should not be used in a provisioning workflow.

As shown in Figure 11-19, the installation output indicates that the log can be checked in `/u01/oraInventory/logs/installActions2012-10-02_05-27-00PM.log`.

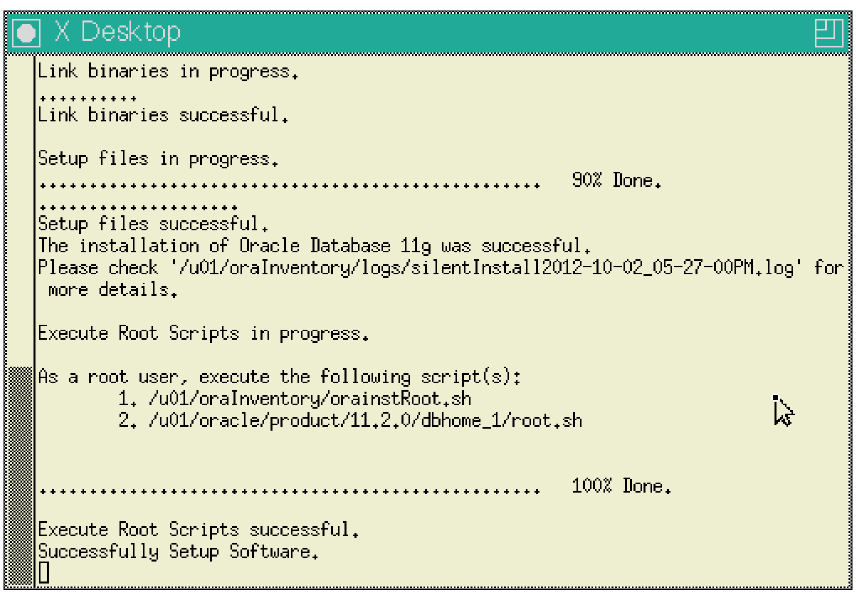


Figure 11-19 Oracle Silent Installation output

2. Look for a correct installation message in the tail of the log file, as shown in the following example:

```
$ tail -10 /u01/oraInventory/logs/installActions<timestamp>.log | grep
"Successfully executed"
```

3. As root user during the post-installation process, run the **orainstRoot.sh** and **root.sh** scripts, which contain all of the product installation actions that require root privileges, as shown on Figure 11-20.

Figure 11-20 Post installation shell output

11.6 Customizing a script

In this section, we describe the script that is used to set the environment and start the Oracle installation on the new provisioned instances.

11.6.1 Customizing the shell script

This script is used by the Tivoli Service Automation Manager provisioning workflow. A sample is provided in “Sample script for preoracle.sh” on page 379.

> **Warning:** The **preoracle.sh** script must be customized according to each environment and then validated before a provisioning tool is used in a production environment.

It implements the following items in the Linux for Oracle environment:

- The first section must be configured with your own environment variables.
- The second section performs the following tasks:
  - Creates groups and users
  - Updates the Oracle profile
  - Copies the silent installation script
  - Tests and creates `/usr/local/bin`, if needed
  - Creates directories that are needed to install Oracle

- Updates kernel parameters in /etc/sysctl.conf
- Updates /etc/limits.conf
- Updates /etc/pam.d/sshd
- Installs rpms for Oracle
- Prepares a vncserver to avoid being prompted when starting vncserver before Oracle installs

- The third section starts Silent runInstaller

The mounted NFS server should contain compressed Oracle code packages. Oracle code is in the /mnt/database directory and custom administration scripts are in the /mnt/script directory, including silent installation and rpms.

Make the script executable by root, as shown in the following example:

```
# chmod u+x preporacle.sh
```

The script can also be started as a stand-alone script, as shown in the following example:

```
# ./preporacle.sh
```

11.6.2 Customizing the Silent Installation file for new server

The previously recorded Silent Installation file contains the host name of the system that is running Oracle, as shown in Example 11-1. This must be updated for each new Oracle instance. The preporacle.sh shell changes the ORACLE_HOSTNAME field inside the Silent Install file

Example 11-1 Response file example with ORACLE_HOSTNAME

```
#-------------------------------------------------------------------------------
# Specify the hostname of the system as set during the install. It can be used
# to force the installation to use an alternative hostname rather than using the
# first hostname found on the system. (e.g., for systems with multiple hostnames
# and network interfaces)
#-------------------------------------------------------------------------------
ORACLE_HOSTNAME=l1oradb1.mop.ibm.com
```

Note: This part is included in preporacle.sh script that is started by the Tivoli Service Automation Manager/Tivoli Provisioning Manager workflow.

11.7 Creating the Tivoli Service Automation Manager and Tivoli Provisioning Manager objects and workflows for PaaS provisioning

This section describes how to implement Tivoli Service Automation Manager objects in a Tivoli environment to deploy a new Oracle Linux instance in z/VM. A custom Oracle Platform as a Service (PaaS) service is available from Tivoli Service Automation Manager Self Service portal.

We also describe the steps that are needed to set up Tivoli Service Automation Manager before it can be used to provision new Linux on System z guests under z/VM. Tivoli Provisioning Manager is the engine that runs workflows on behalf of the Tivoli Service Automation Manager product. The Tivoli Service Automation Manager user interface drives most of its functions through Tivoli Provisioning Manager workflows.

The process includes the following steps:

1. Set up an Oracle Image.
2. Catalog an Oracle Image.
3. Develop the Oracle Installation workflow.
4. Register an Oracle PaaS.
5. Request an Oracle PaaS.

11.7.1 Setting up an Oracle Image

In this section, we describe the steps to set up an Oracle image.

Creating an Oracle Group and users

Create an Oracle Administrator Group and add users to this group.

Cloud and team administrators can create users and add them to the existing teams. Each user is assigned to a customer, security level, one or more security groups, and, optionally, to one or more teams.

In Tivoli Service Automation Manager version 7.2.2, customer objects are introduced as part of the service provider functionality. Because the service provider functionality is not optional and is enabled by default, the existing data must be upgraded to the new model.

Even if you are not planning to use the service provider functionality with Tivoli Service Automation Manager, you must configure at least one customer. For this purpose, Tivoli Service Automation Manager 7.2.2 provides a default customer `PMRDPCUST` of type SP to represent the service provider.

Complete the following steps to create an Oracle Group and add users:

1. In the Service Automation Manager Home panel, select the customer for which you want to create a user, `PMRDPCUST` in our case. Click **Request a New Service → Virtual Server Management → Manage Users and Teams → Create User**.

   The Create User wizard opens, as shown on Figure 11-21 on page 256.

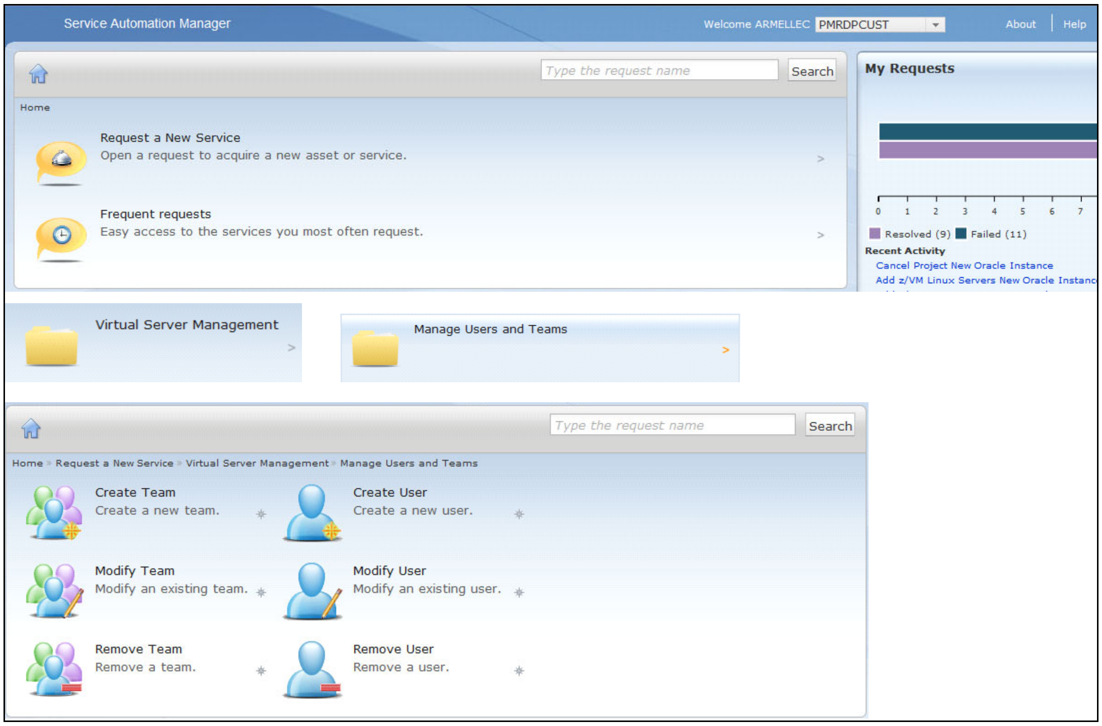


Figure 11-21 Create a user in Tivoli Service Automation Manager

2. Users and Team management can be set in this panel. Create an Oracle team, which includes all Oracle Users, as shown in Figure 11-22.

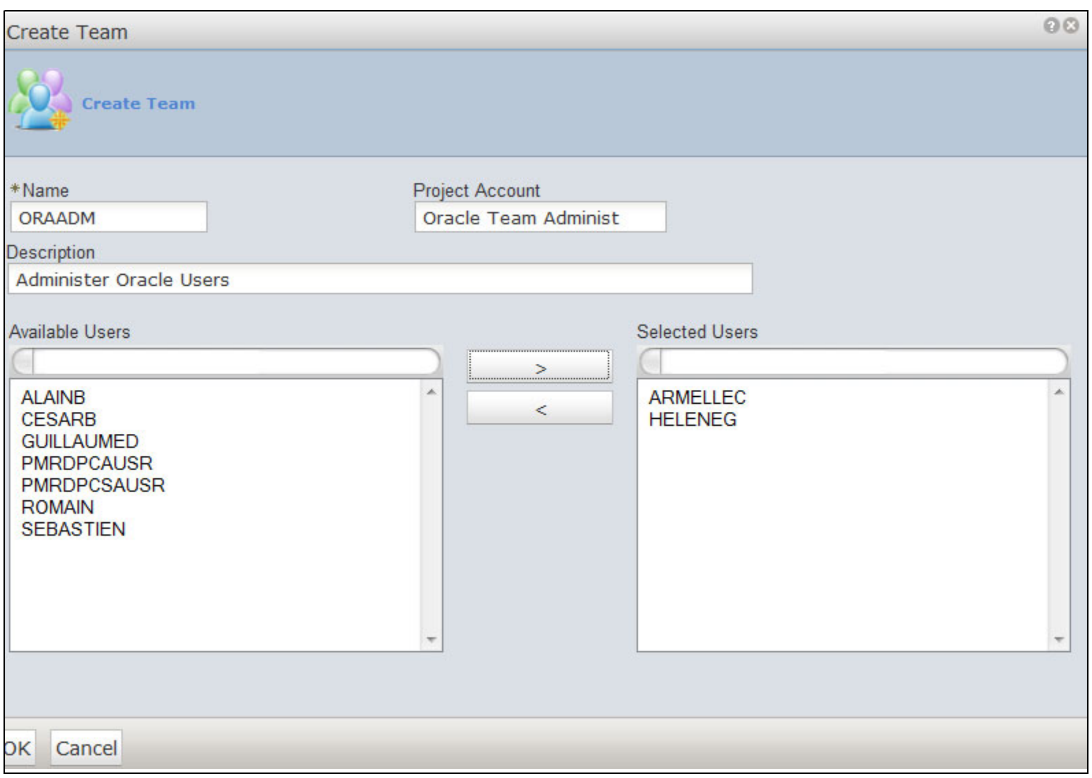


Figure 11-22 Creating a team

3. As shown in Figure 11-23, create other Oracle users that belong to the Oracle Administration team, `ORAADM`, that was created in Step 2 (you can complete the Oracle instances provisioning later). In the Security Group panel, choose **Security Group for Customer Level Policy**.

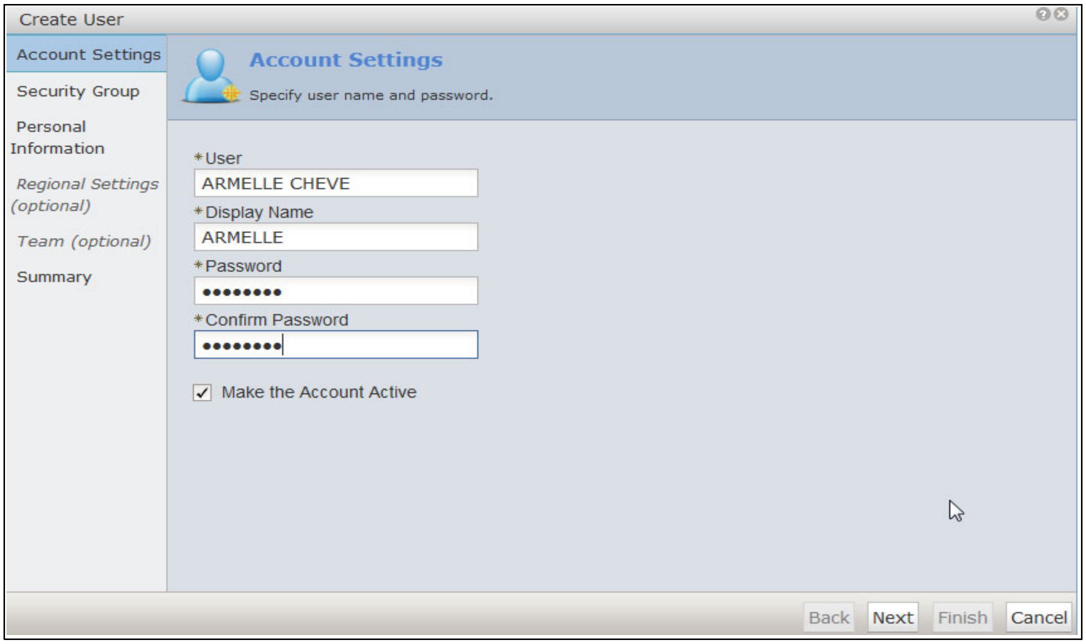


Figure 11-23 Creating users

Configuring resources in Tivoli Service Automation Manager

In this section, we describe the steps to instantiate CPU, Memory, Network, and Disks pools in Tivoli Service Automation Manager. For more information, see Chapter 4 of *Tivoli Service Automation Manager Version 7.2.2 - Installation and Administration Guide,* SC34-2657-00.

The following methods can be used to create the resource pools:

- Create DCM XML files, load the `vrpool.properties` file, perform the discovery, and then validate and enable the cloud server pool.
- Perform all the configuration steps manually by using the Cloud Server Pool Administration application.

The configuration steps that are described next are needed to have resources definition available to Tivoli Service Automation Manager.

Configuring cloud server pools

A cloud server pool is the central object that Tivoli Service Automation Manager uses to define cloud environments. It contains references to all data center model (DCM) resources that are needed for the pool (including hypervisor manager, host platform, file repositories, and resource pools). For more information, see the “Configuring cloud server pools for z/VM” section in the *Tivoli Service Automation Manager Version 7.2.2 - Installation and Administration Guide,* SC34-2657-00.

Configuring cloud networks

Configure your network by completing the following tasks. Create two types of DCM objects, which are necessary for network configuration:

- Sub network: Defines Layer 3 configuration of a network interface within the deployed operating system.
- Switch: Defines Layer 2 connectivity of the network adapter within a virtual machine to the hypervisor virtual switches.

A set of sample DCM import files for hypervisors and resources is provided with Tivoli Service Automation Manager. For more information, see the “Configuring cloud networks” section in the *Tivoli Service Automation Manager Administration Guide*, SC34-2657.

Configuring cloud storage pools

A Tivoli Service Automation Manager Cloud storage pool is a collection of storage resources for more disks. It is associated with a Tivoli Provisioning Manager storage pool.

When creating a project, you can select a server image and set only the size of the local disk. Cloud storage pools are a flexible solution to add other storage to your provisioned servers.

Click **Service Automation** → **Configuration** → **Cloud Storage Pool Administration** and then complete the following steps:

a. Click the **New Cloud Storage Pool** toolbar button and complete the following steps:

   i. Provide the required parameters for the new cloud storage pool.

   ii. Click the **Select Value** icon to display a list of available storage types and storage extension types.

   The default storage type for a logical partition (LPAR) is Mapped Additional Disks. Mounted storage is supported by storage extensions.

b. In the Associated Tivoli Provisioning Manager Storage Pools section, click **New Assignment** to create a storage pool table entry. Select one Tivoli Provisioning Manager storage allocation pool object.

> **Storage discovery:** If there are no storage allocation pools listed in the table, storage discovery likely was not run. To run storage discovery, open the Cloud Server Pool Administration application, select an available System z cloud pool, and run the discovery from the Storage Discovery tab.

c. Switch to the Security Settings tab and specify the required information.

d. Click **Validate and Enable Cloud Storage Pool**.

For more information, the “Configuring cloud storage resources” section in the *Tivoli Service Automation Manager Administration Guide*.

Configuring the service provider and customer features

Even if you are not planning to use the service provider functionality, you must perform the following basic configuration steps to make the default customer operational.

From the Start Center main page, click **Go to** → **Service Automation** → **Configuration** → **Cloud Customer Administration**.

To register the default Cloud Service Provider (`PMRDPCUST`), click the **New Customer** icon, as shown in Figure 11-24 on page 259.

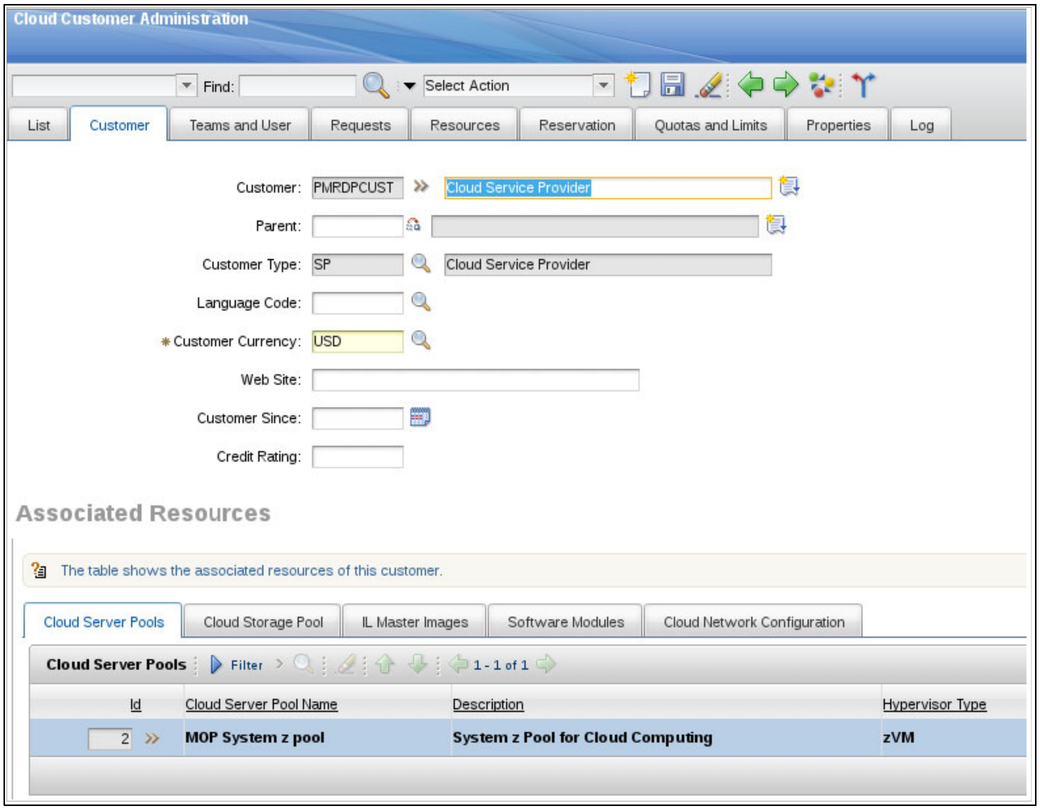


Figure 11-24 Customer definition

A customer is associated with one or more Cloud Server Pools. Assign Resources Pools that were previously created to the new Cloud Provider.

Tivoli Service Automation Manager maintains a large collection of objects and their attributes, which forms its DCM that is stored in a separate database. Each object in Tivoli Service Automation Manager Data model is assigned a unique identification number. The objects description can be retrieved from the Data Model Object Finder, which is in the main page of the Tivoli Service Automation Manager Start Center.

Objects defined in our testing environment

We defined the following objects in our testing environment:

- Cloud Server Pools

  The Cloud Server pool is represented by a DCM object named `MAPSRV`. It is a logical object that represents the z/VM partition and maintains the status of all its available resources: CPU, memory, and disk. Figure 11-25 shows information about the Cloud Server Pool that is defined in our environment.

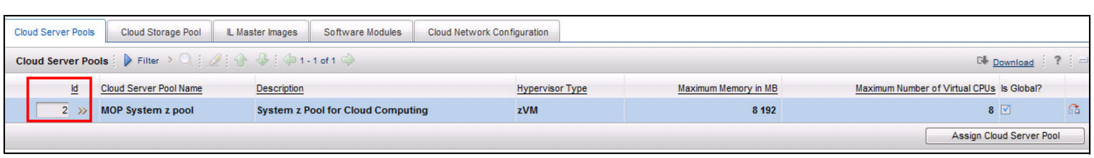


Figure 11-25 Cloud Server Pool definition

> **Cloud Storage Pools:** The Cloud Storage Pools are not used in our case. Instead, we use the `MAPSRV` server default pool. A z/VM discovery gathers disk information about the default server pool `MAPSRV` (no storage pool is required).

- Master Images, as shown in Figure 11-26:

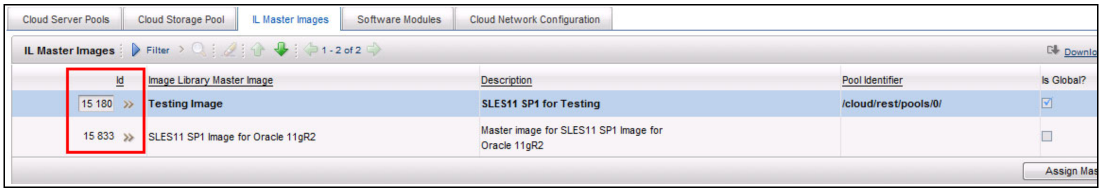


Figure 11-26 Master Image Library

- Software Stack, as shown in Figure 11-27:

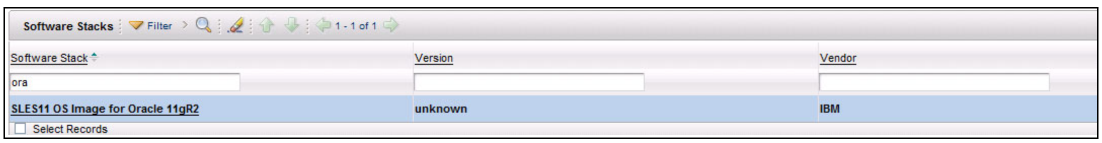


Figure 11-27 Software Stack

- Cloud Network Configuration, as shown in Figure 11-28:

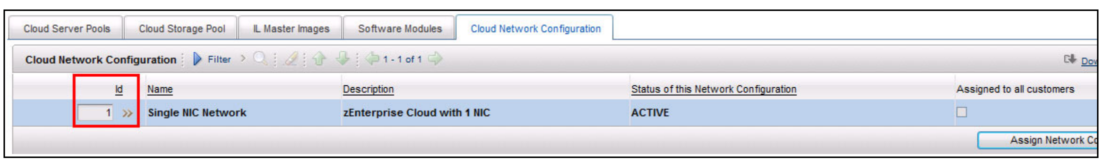


Figure 11-28 Cloud Network configuration

For more information, see the “Configuring the service provider” section of the *Tivoli Service Automation Manager Administration Guide*.

Configuring the XML files that are imported into the DCM

This step describes the process that is used to tailor the XML files that are used to import requested objects into Tivoli Service Automation Manager. This process includes the following overall steps:

1. Tivoli Service Automation Manager distributes sample XML files with the product.
2. These files must be tailored to the local Linux and z/VM environment.
3. The resulting XML input files are incorporated into the Tivoli Service Automation Manager database and the DCM.
4. Any text editor (for example, Notepad) can be used to work with `.xml` files.
5. All objects that are contained in the `.xml` file are imported into the DCM.

To build an Oracle image and use it with the default Tivoli Service Automation Manager service for Linux that is running as a guest under z/VM, we prepared and built the following elements in a Tivoli Service Automation Manager environment:

- Oracle Master Image
- Software Stack
- Virtual Server Template

  We need to register a new image in the Tivoli Service Automation Manager image library for the Linux under z/VM. To instantiate new Oracle environments, we must create a Tivoli Provisioning Manager workflow. This workflow is started after the Linux OS installation in the virtual machine, as described in “Updating the Software Stack” on page 265.

  Tivoli Service Automation Manager provides a way to directly define DCM Objects in its interface or to build `.xml` files and import them by using the administrative interface to import the following components:

  – Software Stack
  – Master Image
  – Virtual Server Template

- Master Image Template: OraclezLinuxImageSLES11SP1_zVMMOP2012.xml

  To have a Master Image available, the virtual server template must have some information in it, such as, Linux Prototype, Linux reference name, and the MDisk that is to be cloned, as shown in Figure 11-29.

```
   <image name="SLES11 SP1 for Oracle" image-type="Golden_Master
description="Prepared for TSAM" locale="en_US" version="1.0"
boot-server="MAPSRV-bootserver" status="tested" is-device-
model="SOAonRAMPimage" software-module="SLES11 GM OS for Oracle
priority="1">
    <property name="zVM_Prototype" component="KANAHA"
value="LNXPROTO"/>
    <property name="zVM_DiskOwnerId" component="KANAHA"
value="L1ORAREF"/>
    <property name="zVM_CloneDisks" component="KANAHA"
value="100"/>
    <property name="zVM_SystemDisk" component="KANAHA"
value="100"/>
    <property component="KANAHA" name="cp_monitor" value="true"/
    <property component="KANAHA" name="cp_swap" value="false"/>
    <property component="KANAHA" name="swType" value="OS"/>
  </image>
```

Figure 11-29 Extract from the XML Master Image, OraclezLinuxImageSLES11SP1_zVMMOP2012.xml

- Virtual Server Template: `VirtualServerTemplateSLES11_zVMMOP2012.xml`, as shown in Figure 11-30 on page 262.

```
  <!-- Define all involved zLinux images -->
  <!-- software stack is a software module containing software
module(s) or images -->
    <virtual-server-template name="ZNTC VST - 1 NIC - 2 CPUs -
1gb Storage" image-stack-name="SLES11 GM OS for Minidisks">
    <resource-requirement resource-type="platform" how-many="0"
size="0.0" is-shared="true"/>
    <resource-requirement resource-type="cpu" how-many="2"
size="0.0" is-shared="true">
      <property component="KANAHA" name="cpu.family"
value="s390"/>
    </resource-requirement>
    <resource-requirement resource-type="memory" how-many="1024"
size="0.0" is-shared="true"/>
    <resource-requirement resource-type="disk" how-many="1"
size="6.87812805" is-shared="false">
      <property component="KANAHA" name="zVM_DiskDeviceNumber"
value="100"/>
      <property component="KANAHA" name="zVM_DiskType"
value="Minidisk"/>
    </resource-requirement>
    <property component="KANAHA" name="zVM_Prototype"
value="LNXPROTO" description="Optional - initial z/VM prototype
file used to create a virtual server"/>
    <property component="KANAHA" name="host-platform-type"
value="zVM"/>
    <property component="KANAHA" name="zVM_Userid"
value="dummy"/>
  </virtual-server-template>
```

Figure 11-30 Extract from Virtual Server Template XML sample

11.7.2 Cataloging an Oracle Image

By completing this task, the needed definitions in Oracle environment can be defined in Tivoli Service Automation Manager DCM.

Importing resources

This step consists of importing the `.xml` files that were created in “Configuring the XML files that are imported into the DCM” on page 260 into the DCM. This task is done by the Tivoli Service Automation Manager Administrator.

From the Start Center main page, click: **Go To** → **Service Automation** → **Configuration** → **Cloud Server Pool Administration**, as shown on Figure 11-31 on page 263.

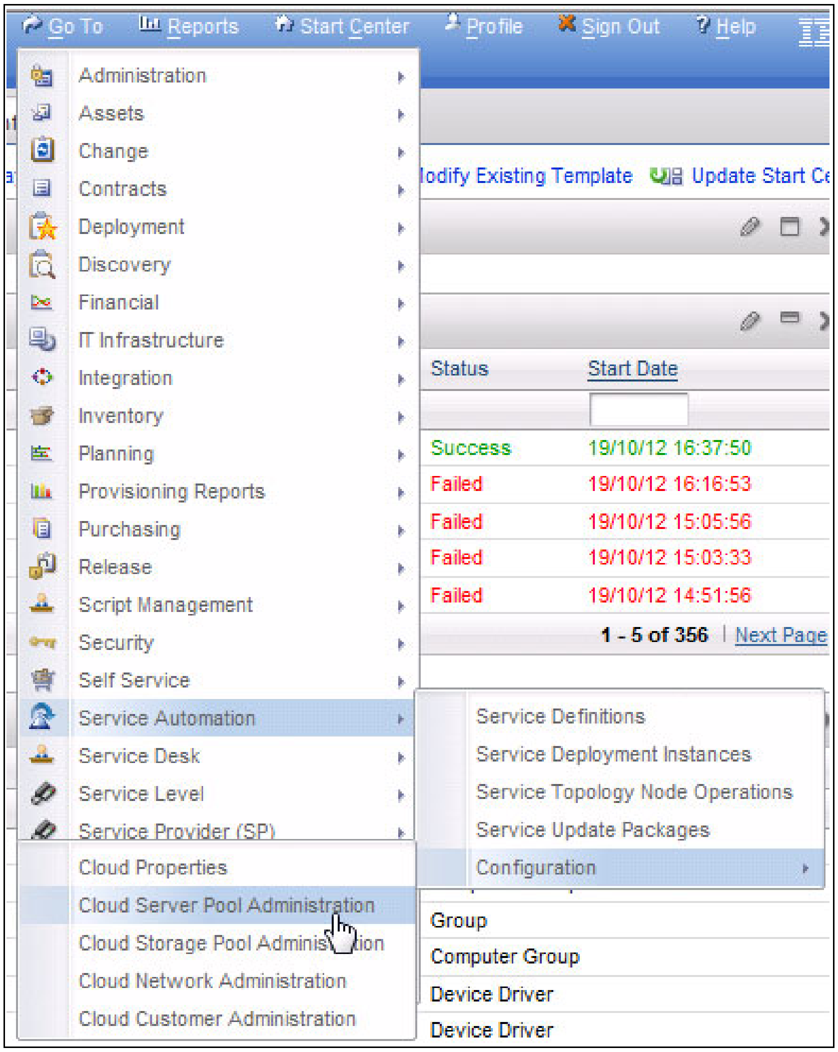


Figure 11-31 Cloud Server Pool administration access from Go To menu

Click **Import DCM Objects**. This import creates the DCM objects in the Tivoli Service Automation Manager's database. We use those objects to register our new Oracle Image. This is the image we select with the default service of Tivoli Service Automation Manager Create Project Linux under z/VM.

Displaying the Oracle Master Image

This step displays the Master Image for Oracle instances.

The Master Image can be created by using the importing an `.xml` file process or manually through the Tivoli Service Automation Manager administration GUI. The `.xml` file that is imported into the DCM first must be configured by using a text editor (for example, Notepad).

To display the Master Image for Oracle, from Start Center main page, click: **Go To** → **IT Infrastructure** → **Image Library** → **Master Images**, then provide Search argument *`SLES11 SP1 Image for Oracle 11gR2`*. (The search result is shown on Figure 11-32 on page 264). Click the name of the master image to see its details and note how it is linked to the Virtual Server Template and Software Stack.

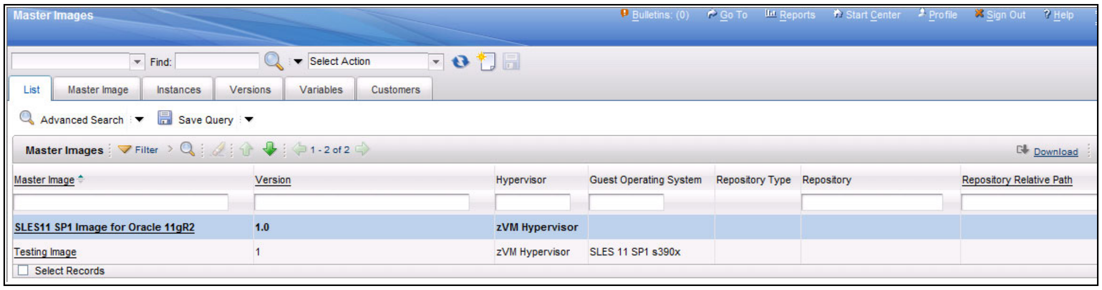


Figure 11-32 Searching for Master Images

Displaying the Virtual Server Template

To display the Virtual server Template (VST) content, click: **Go To** → **IT Infrastructure** → **Provisioning Inventory** → **Virtual Server Templates**, as shown in Figure 11-33.

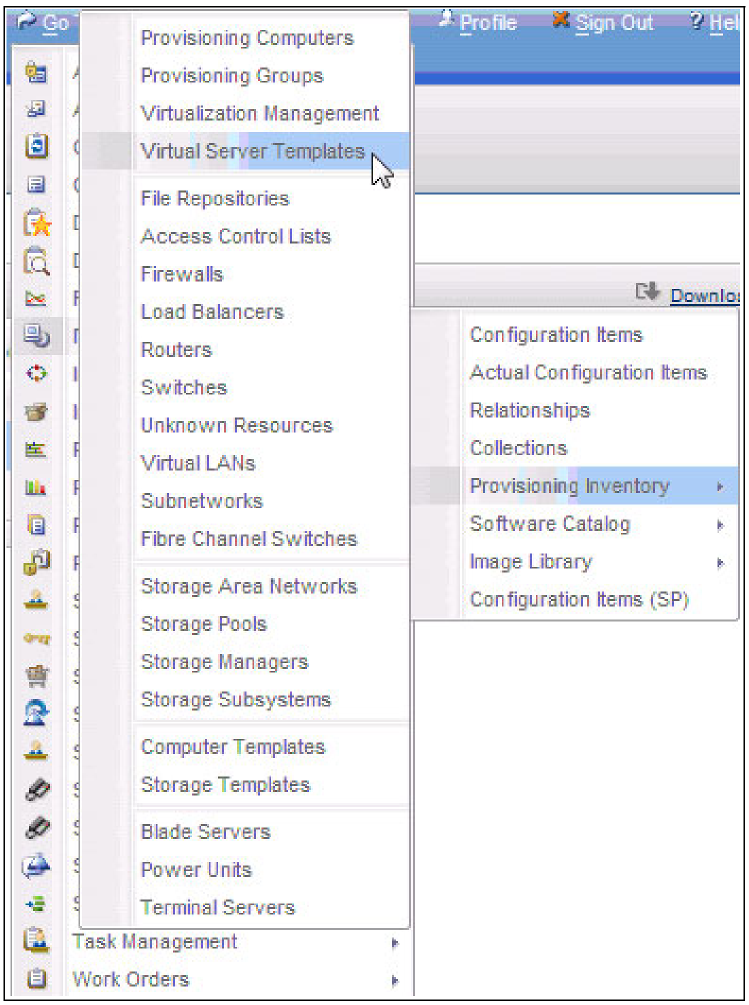


Figure 11-33 Virtual Templates administration access from GoTo menu

Search for keyword `Oracle` in the Virtual Server Templates search field. The results are shown in Figure 11-34.

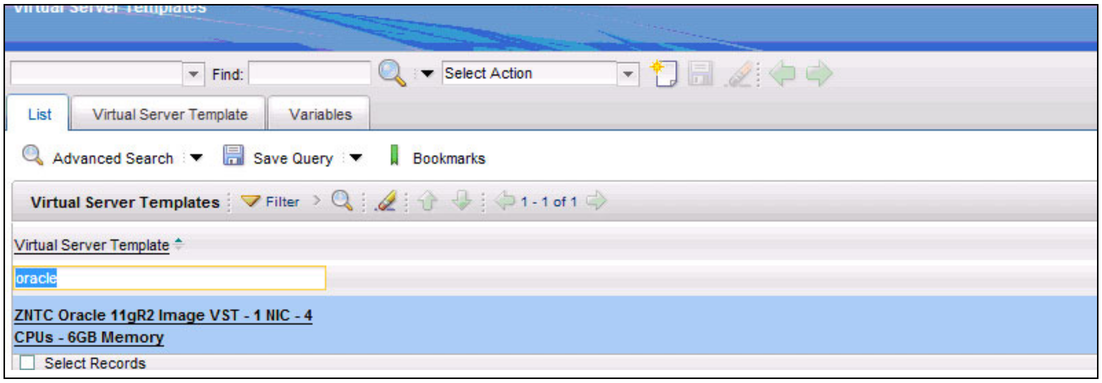


Figure 11-34 Searching for Virtual Server Templates

Click the VST name to display its content.

Updating the Software Stack

This step consists of adding a software module (`Oracle11gR2`), to the Software Stack `SLES11 SP1`, which is defined by `SLES11 OS Image for Oracle11gR2`.

Oracle11gR2 software module was previously modified to call our homemade workflow, which is called `Oracle11203_Install`.

For more information about designing the workflow, see “Sample workflow for Linux and Oracle installation” on page 268.

To add a software module (Oracle11gR2) to the Software Stack, open the software stack document by clicking **Go To** → **IT Infrastructure** → **Software Catalog** → **Software Stacks** and search for keyword “Oracle” in the Software Stack field.

Select the software stack, `SLES11 OS Image for Oracle 11gR2` in our example, as shown in Figure 11-35.

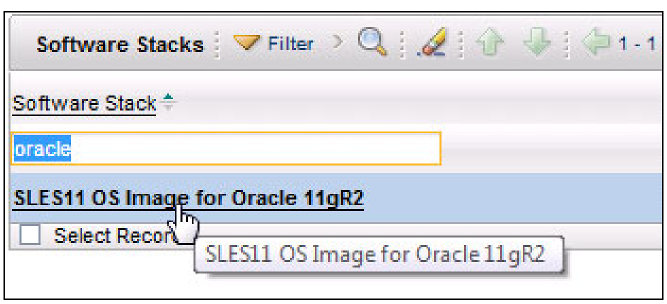


Figure 11-35 Selecting a Software Stack.

This displays the SLES11 for Oracle Software Stack. In the Select Action box, select **Add Stack Entry**.

In the Add Stack Entry panel (as shown on Figure 11-36 on page 266), search for Software definitions that contain Oracle, and then select the Oracle definition that contains your workflow. For more information about designing the workflow, see “Sample workflow for Linux and Oracle installation” on page 268.

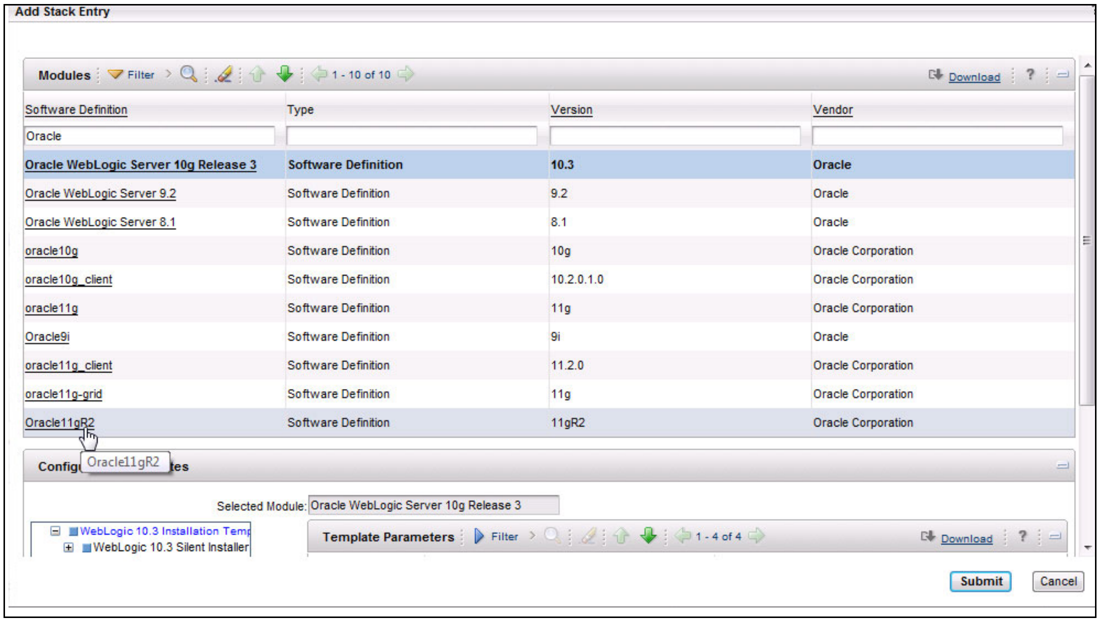


Figure 11-36 Add Stack Entry panel

Select the Oracle chosen template and click **Submit**. The Oracle software (workflow) is added to the Software Installable section of the Software stack configuration for Oracle provisioning.

Save the new Software Stack configuration. Click the **Save** icon. The Oracle 11gR2 Software Installable module is added to the Linux SLES11 OS Image for Oracle 11gR2. This results in a new software module that can install Oracle software immediately after the new Linux Instance is generated in a single project. The `oracle11g_SoftwareInstallable_Install` workflow is now associated with this Software Stack because you can check in the Software Products window.

Creating a Software Installable Post Install workflow

This step consists of adding a new workflow for Post Installation actions, then assigning it to the SLES11 SP1 Software Stack, which is defined by the SLES11 OS Image for Oracle11gR2 Software Stack object.

First, we need to define a new Post Install workflow. To help in this task, we copy an existing workflow and create one that is dedicated to Oracle post installation.

Complete the following steps:

1. Go to the Start Center portal main page. In Automation Development Applications, click **Provisioning Workflow** and enter `InstallPost` in the Workflow Name field, as shown in Figure 11-37 on page 267.

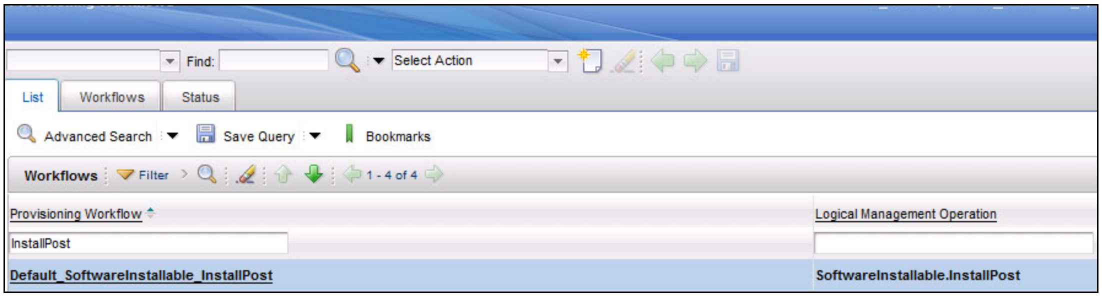


Figure 11-37 Edit a Post Install workflow source code

2. Click the link to edit the code and then copy it into the clipboard.
3. Complete the following steps to create a Post Install workflow for Oracle:
   a. Click the **New Workflow** icon that is the top of the page.
   b. Paste the code that was previously inserted in the clipboard, and then save the new workflow with a new name: `Oracle_SoftwareInstallable_InstallPost`, as shown in Figure 11-38.

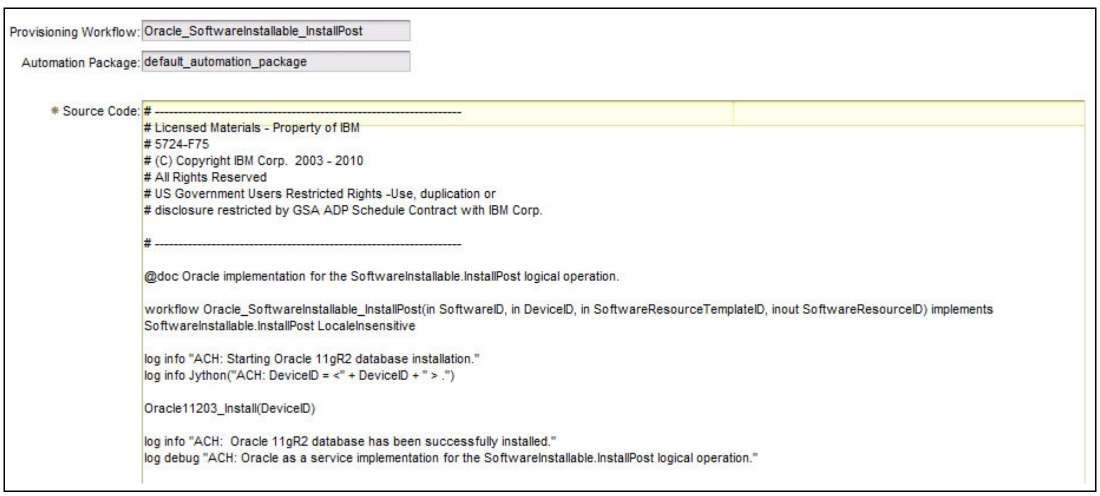


Figure 11-38 Create a Post Install workflow for Oracle

   c. Add the Post Installation workflow, as described in “Oracle Post Install actions” on page 268. Debug the information to facilitate problem analysis.
   d. Click the **Save** icon at the top of the page. The new code is compiled. Click **OK** when completed.
4. Add the workflow to the Oracle 11gR2 software stack. Open the software stack as described in “Updating the Software Stack” on page 265, and shown in Figure 11-35 on page 265.
5. From the Software Stack tab, select the **Workflows** tab in the Installable Files section, and click **Assign Provisioning Workflow**.
6. In the Assign Workflow list, search for the InstallPost available workflows, as shown in Figure 11-39 on page 268.

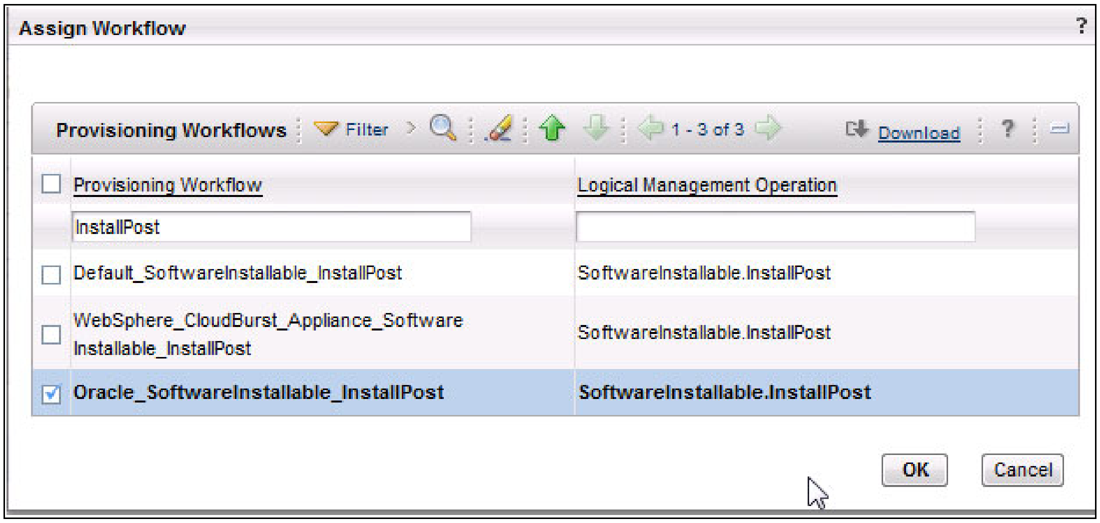


Figure 11-39 Assign a Post Install workflow.

7. Select the Oracle post installation workflow and click **OK**, then agree to save the changes.

   The new post installation workflow is added to the Software Installable files.

Oracle Post Install actions

Two Oracle Installation process scripts (`orainstRoot.sh` and `root.sh`) must be ran after the Oracle installation process is complete. The scripts contain the product installation actions that require root privileges.

For more information about these scripts, see “Replaying the installation in Silent Mode” on page 252.

11.7.3 Creating a Workflow

Different Tivoli Provisioning Manager workflows can be developed. In our case, we developed one workflow for Oracle Installation.

Sample workflow for Linux and Oracle installation

The Tivoli Provisioning Manager deployment engine is responsible for creating, storing, and running repeatable workflows that automate the configuration and allocation of IT assets. A workflow can represent an entire reconfiguration process that affects multiple computers or a single step in a larger reconfiguration process.

A workflow is represented by a script that uses the following objects:

- Workflow keyword that identifies a file as a workflow
- Variables and assignment
- Calling workflows and device operation, which help to reuse workflows in different contexts
- Calling Java methods
- Scriptlets, which are small scripts included in the workflow
- Jython and operators that use operators to perform simple logic and handle text strings
- Localization and globalization commands
- Conditional statements, Iterations, annotations, and comments
- Data model queries
- Errors and messages
- Special characters and reserved words

An external script is useful when the script is long and easier to maintain separately, or when the script is used by multiple workflows or workflows in different automation packages.

An external script file (called `preporacle.sh`)is used in this chapter and performs the following tasks:

1. Uses the `Device.CopyFile` device operation to copy the script to the target computer. First, the script is manually uploaded to the Tivoli Provisioning Manager server in a repository directory. Then, the `Device.CopyFile` function copies the script from Tivoli Provisioning Manager repository to the new provisioned guest.
2. Runs the shell script by using the **`Device.ExecuteCommand`** operation.

The following functions parameters are available:

- **`Device.CopyFile`**, `SourceDeviceID`, `SourcePath`, `SourceFile`, `DestinationDeviceID`, `DestinationPath`, `DestinationFile`, `ClientCredentialsKey`, `HostCredentialsKey`, and `TimeoutInSeconds`
- **`Device.ExecuteCommand`**, `DeviceId ExecuteCommand`, `WorkingDirectory`, `CredentialsKey`, `TimeoutInSeconds`, `TreatTimeoutAs`, `ReturnCode`, `ReturnErrorString`, and `ReturnResult`

For more information about the Workflow design process, see the Tivoli Information Center at this website:

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/index.jsp?topic=%2Fcom.ibm.tsam.doc_7.2%2Ft_config_zvm_setup.htmlhttp://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/index.jsp?topic=%2Fcom.ibm.tsam.doc_7.2%2Ft_config_zvm_setup.html

Starting the workload

In this section, we describe how to install Oracle code with a workflow and an external shell script on a test server. This is needed to deploy and test your code before integrating it into the Oracle Software Stack. Complete the following steps:

1. Upload your customized shell script to the Tivoli Provisioning Manager server in a repository directory (here, it is `/opt/ibm/oracle`) and then change the permission bits to allow running with **`chmod u+x preporacle.sh`** command.
2. Select **Provisioning Workflows** in the Automation Development Applications section.
3. To create a workflow, click the **New Workflow** button, as shown on Figure 11-40.

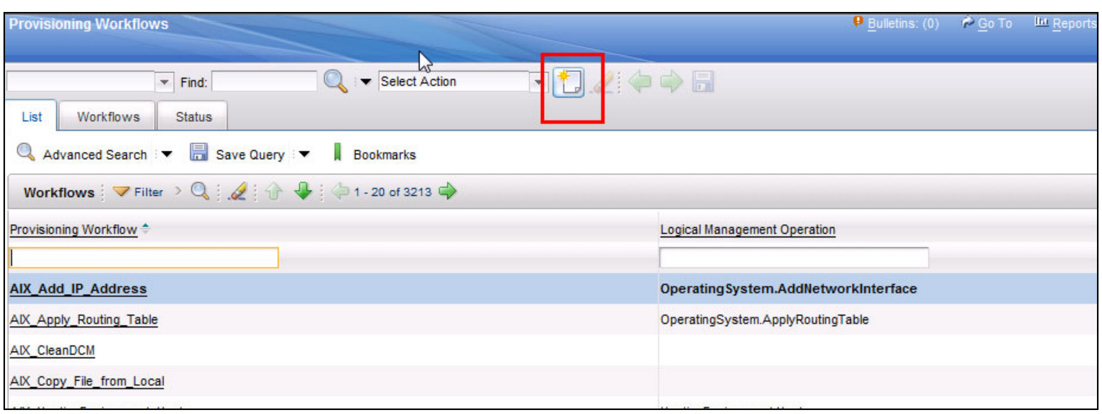


Figure 11-40 Adding a new provisioning workflow

4. A new workflow editor is displayed in the browser. Copy and paste your code inside the Workflow editor, as shown on Figure 11-41. Click the **Save i**con. The new code is compiled.

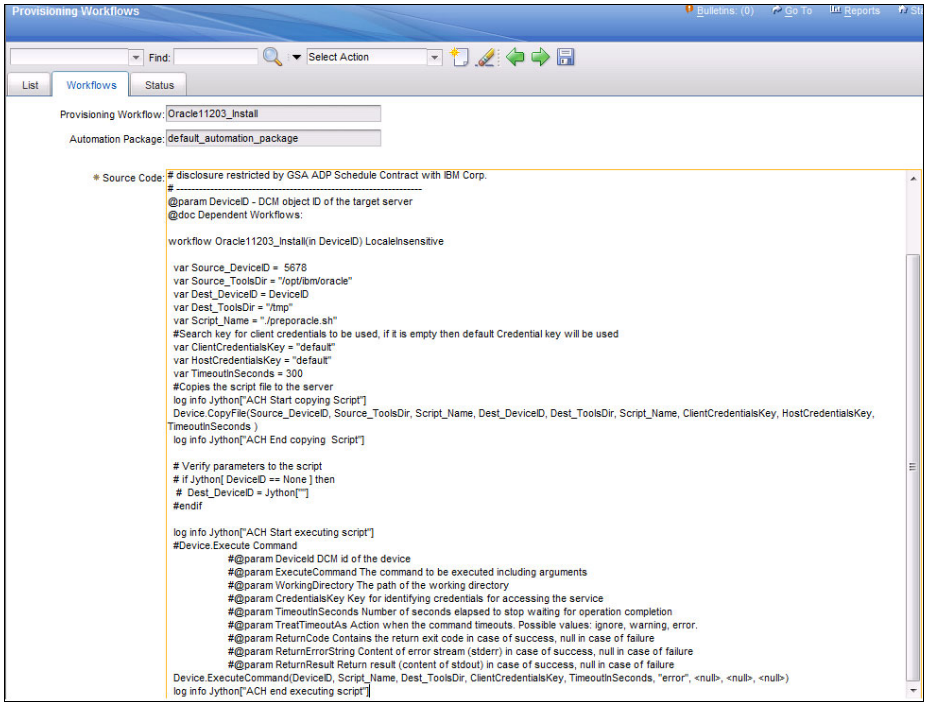


Figure 11-41 Sample script

Click **OK** when completed, as shown in Figure 11-42. Return to the workflow list with the List tab and select the new workflow.

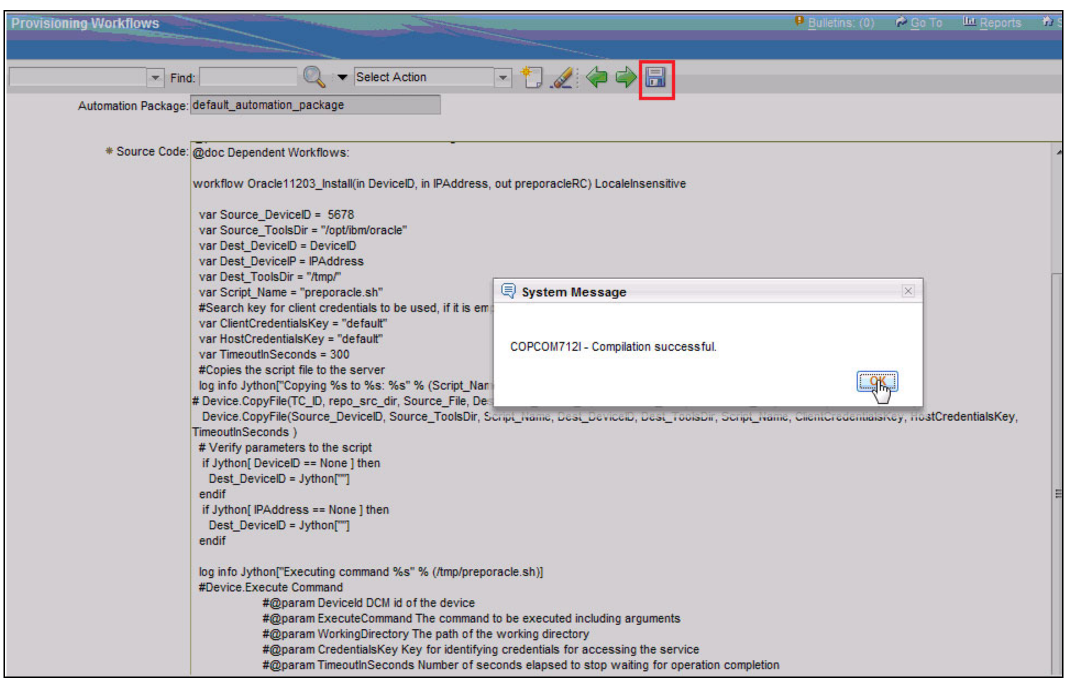


Figure 11-42 New code compilation

5. Click the green arrow to start the workflow, as shown in Figure 11-43.

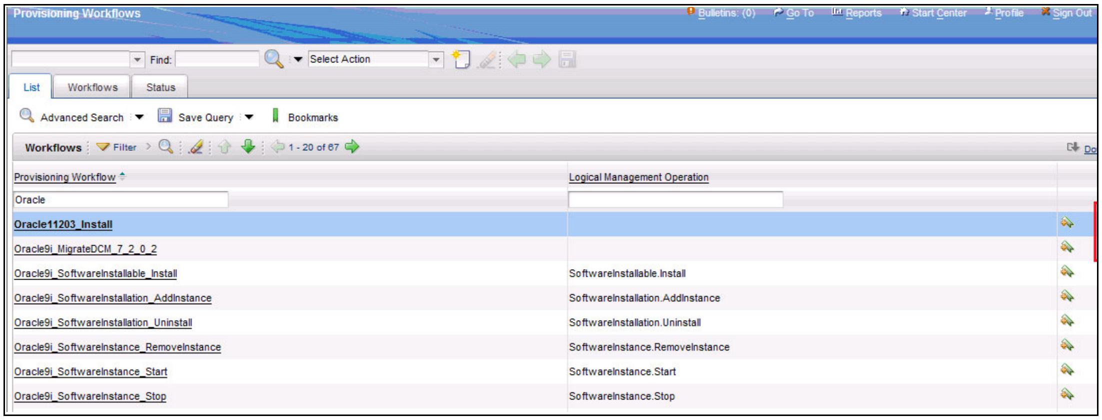


Figure 11-43   Run the new workflow

6. Enter the requested parameters in the workflow panel, as shown in Figure 11-44. Here, the requested parameter is the ID of a Linux Instance where Oracle code is installed. Click **Run** and then click **Yes** to open the Provisioning Task Tracking window.

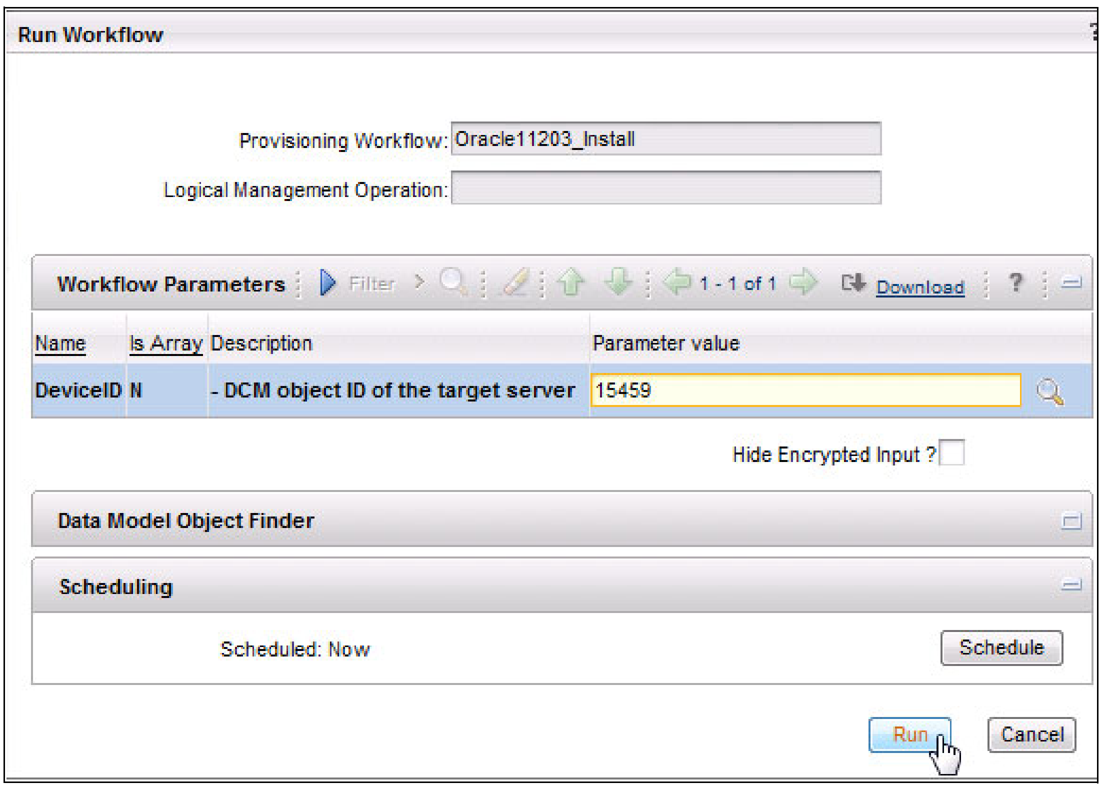


Figure 11-44   Enter parameters panel

7. Click the Workflow Log ID to show the Provisioning Task Tracking log, as shown in Figure 11-45.

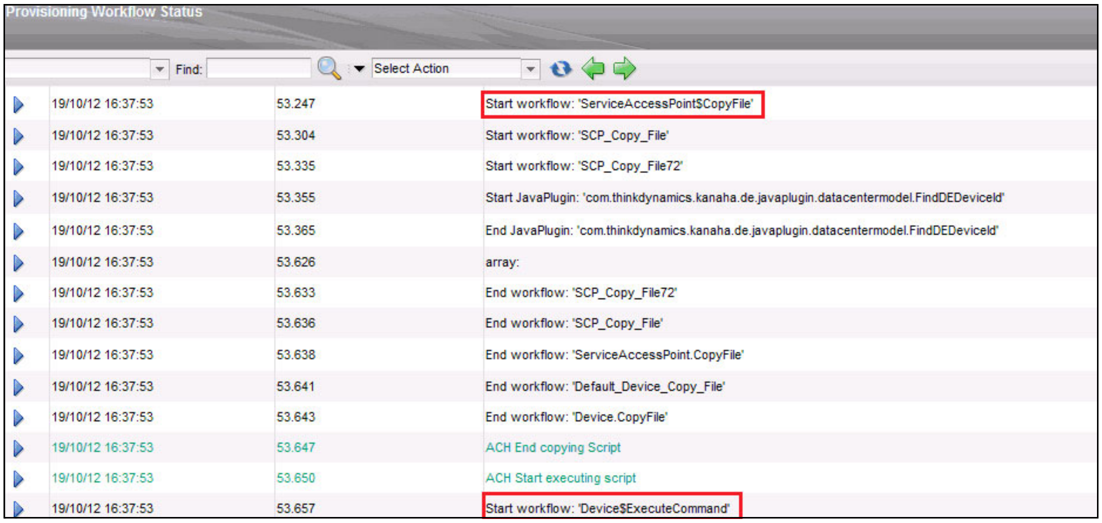


Figure 11-45 Workflow log

The Oracle installation on the new Instance, which is started by the Tivoli Provisioning Manager server, can also be monitored by using the following methods from the new instance environment:

- List the `/mnt` directory files (mounted by the preporacle script).
- List the temporary files directory `/tmp` and the created Oracle logs.
- List the Oracle installation process by using the **`ps -fu oracle`** command.
- Monitor the system performance that shows Java and I/O activity by using the **`top`** or equivalent command.

8. After workflow successful completion, the Status is shown in the Workflow Execution Logs tab, as shown in Figure 11-46.

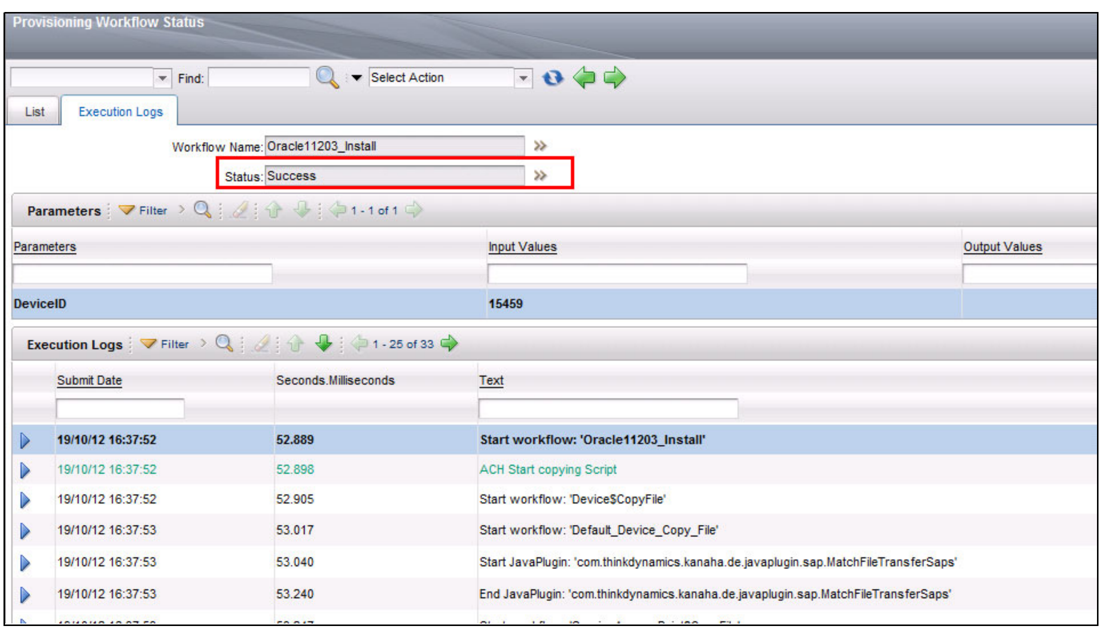


Figure 11-46 Workflow completion status

11.7.4 Registering an Oracle PaaS

In this section, we describe the process that is used to register an Oracle PaaS.

Registering a new z/VM Project in Tivoli Service Automation Manager

This step consists of registering of a new provisioning service, which is done by the Oracle Group Administrator. In a browser, connect to the Tivoli Service Automation Manager Self Service portal main page at this website:

https://l1ntccloud.mop.fr.ibm.com:9443/SimpleSRM/login.jsp

Complete the following steps from the Register z/VM Image window:

1. Provide your User and Password and click **Logon**.
2. Click **Request a New Service**.
3. Click **Virtual Server Management**.
4. On the Virtual Server Management page, click **Manage Image Library**.
5. On the Manage Image Library page, click **Register z/VM Image**.
6. On the Register z/VM Image → General panel, complete the fields, as shown in Figure 11-47. The Cloud Server Pool is the resource pool that is used for provisioning. Choose the Master Image for Oracle in the Discovered Image drop-down menu. Click **Next**.

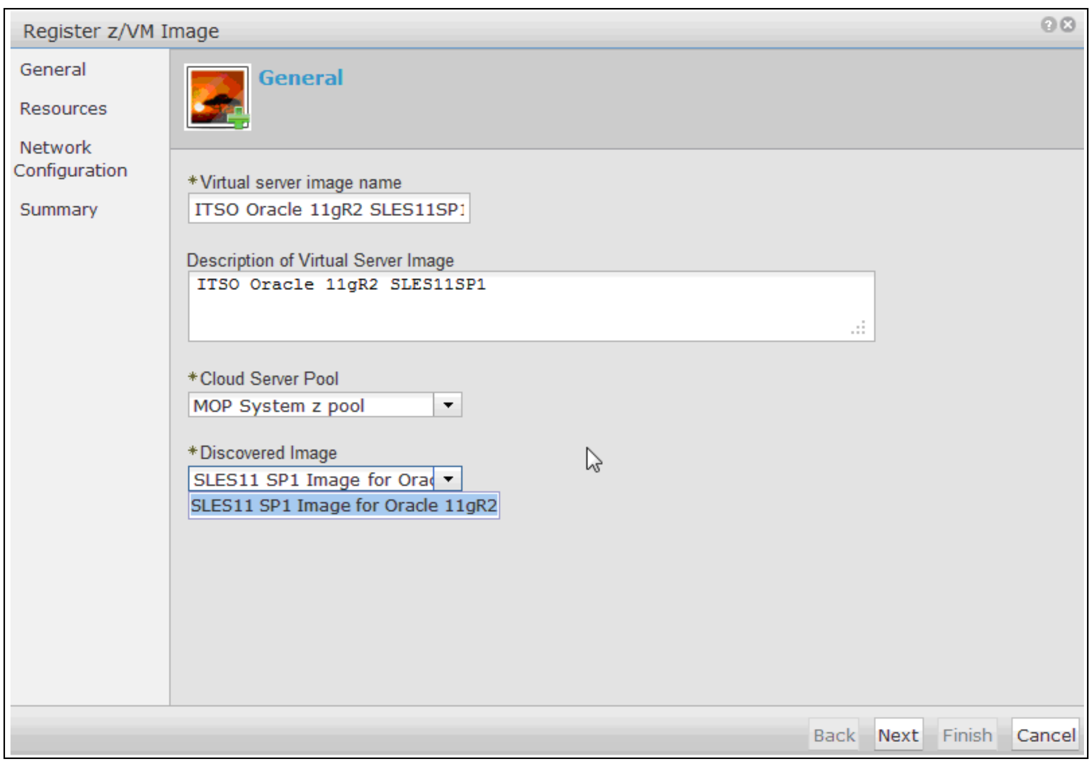


Figure 11-47 Register z/VM Image

7. Adjust the values in the Resources window (if needed) and then click **Next**.
8. Make any change (if needed) in Network Configuration window and click **Next**.
9. A Summary window opens. Click **Finish**.

Follow the progress in the My Request portlet, as shown in Figure 11-48.

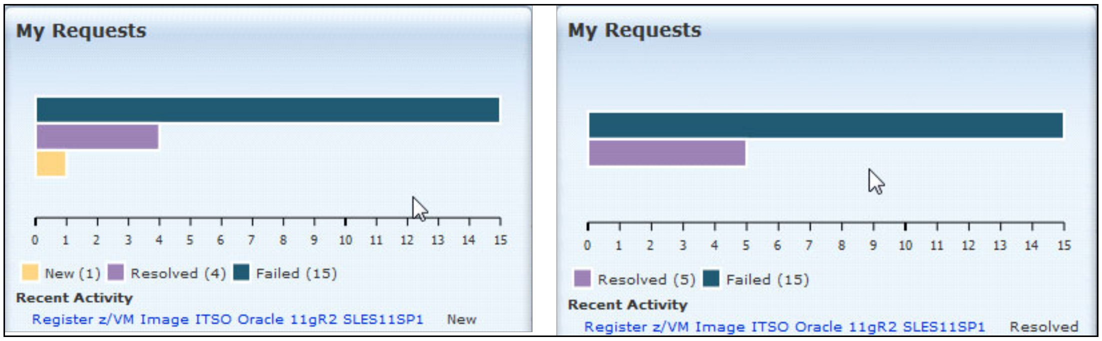


Figure 11-48 Follow the request progress

11.7.5 Requesting a new Oracle Service

In a browser, connect to the Tivoli Service Automation Manager Self Service portal main page and complete the following steps:

1. Click **Request a New Service**.
2. Click **Virtual Server Management**.
3. In the Virtual Server Management window, click **Create Project with z/VM Linux Servers**.
4. Complete the following panels to define the provisioned instances:
   – Project Details, as shown in Figure 11-49
   – Requested Image
   – Servers Details (optional)
   – Additional software (optional)
   – Network Configuration (optional)
   – Other Settings (optional)
   – Summary

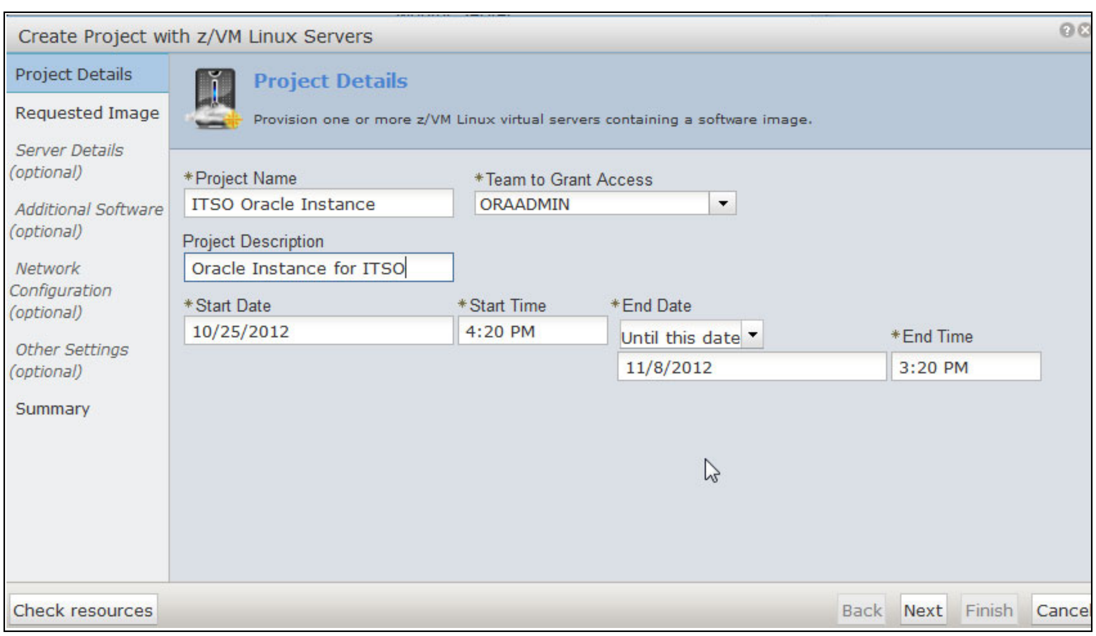


Figure 11-49 Create project with z/VM Linux Servers

Tivoli Service Automation Manager application workflow sends an email showing the approval of the service to the requester, as shown in Figure 11-50.

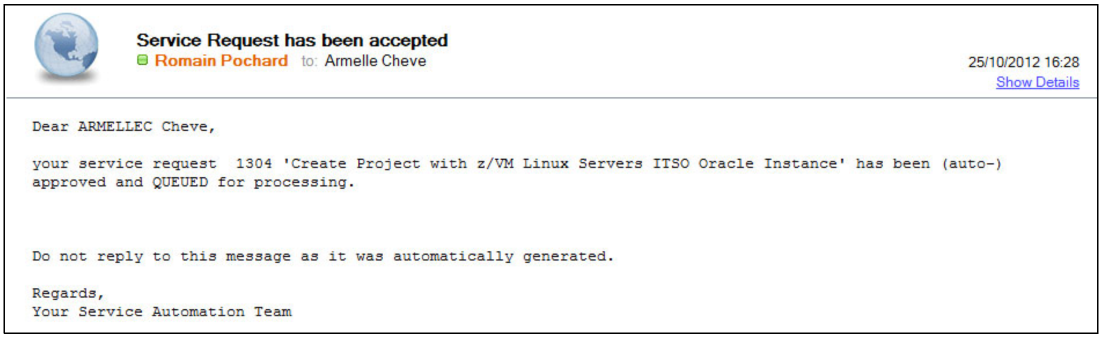
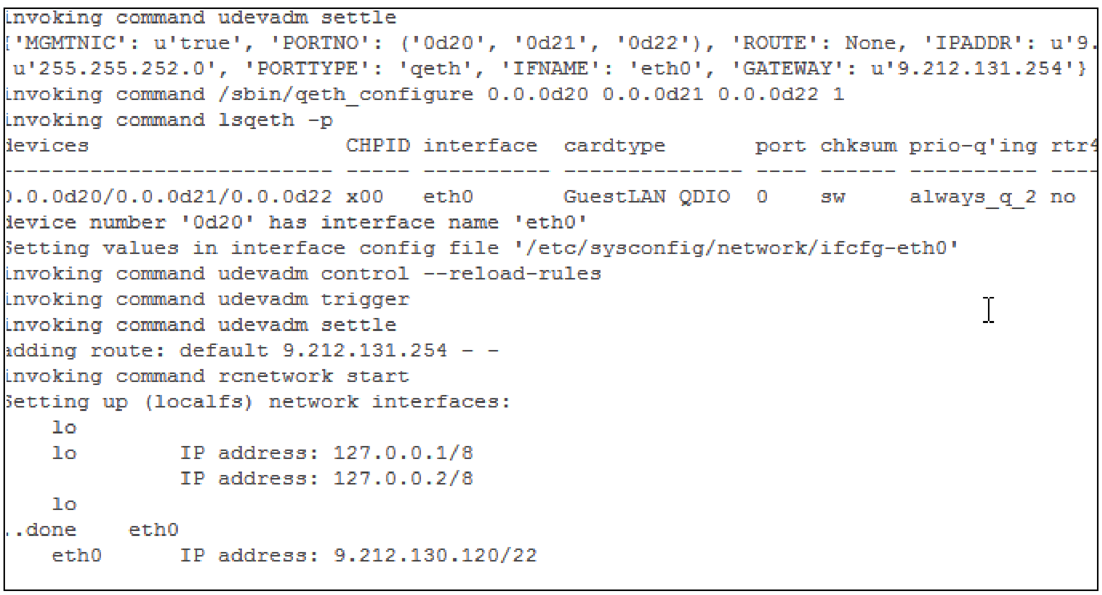


Figure 11-50 Service request approval mail

5. Follow the progress in My Request or My Projects portlets.

   The progress can be followed at `MAPSRV` server log level. Open an SSH session on `MAPSRV` and go to `/opt/ibm/ztsam/workflow/tmp`. Follow the instantiation progression by looking at the end of the current (most recent) log file in the log directory as shown in Figure 11-51:

```
$ cd /opt/ibm/ztsam/workflow/tmp
$ tail -f  logfile.xxxxx
```



```
invoking command udevadm settle
{'MGMTNIC': u'true', 'PORTNO': ('0d20', '0d21', '0d22'), 'ROUTE': None, 'IPADDR': u'9.
 u'255.255.252.0', 'PORTTYPE': 'qeth', 'IFNAME': 'eth0', 'GATEWAY': u'9.212.131.254'}
invoking command /sbin/qeth_configure 0.0.0d20 0.0.0d21 0.0.0d22 1
invoking command lsqeth -p
devices                    CHPID interface  cardtype       port chksum prio-q'ing rtr4
-------------------------- ----- ---------- -------------- ---- ------ ---------- ----
0.0.0d20/0.0.0d21/0.0.0d22 x00   eth0       GuestLAN QDIO  0    sw     always_q_2 no
device number '0d20' has interface name 'eth0'
Setting values in interface config file '/etc/sysconfig/network/ifcfg-eth0'
invoking command udevadm control --reload-rules
invoking command udevadm trigger
invoking command udevadm settle
adding route: default 9.212.131.254 - -
invoking command rcnetwork start
Setting up (localfs) network interfaces:
    lo
    lo        IP address: 127.0.0.1/8
              IP address: 127.0.0.2/8
    lo
..done    eth0
    eth0      IP address: 9.212.130.120/22
```

Figure 11-51 MAPSRV log file extraction

Progress can also be tracked from Start Center Welcome page.

11.7.6 Suppressing a provisioning project

To suppress the provisioned project, click **Cancel t**in the self-service interface, as shown in Figure 11-52.

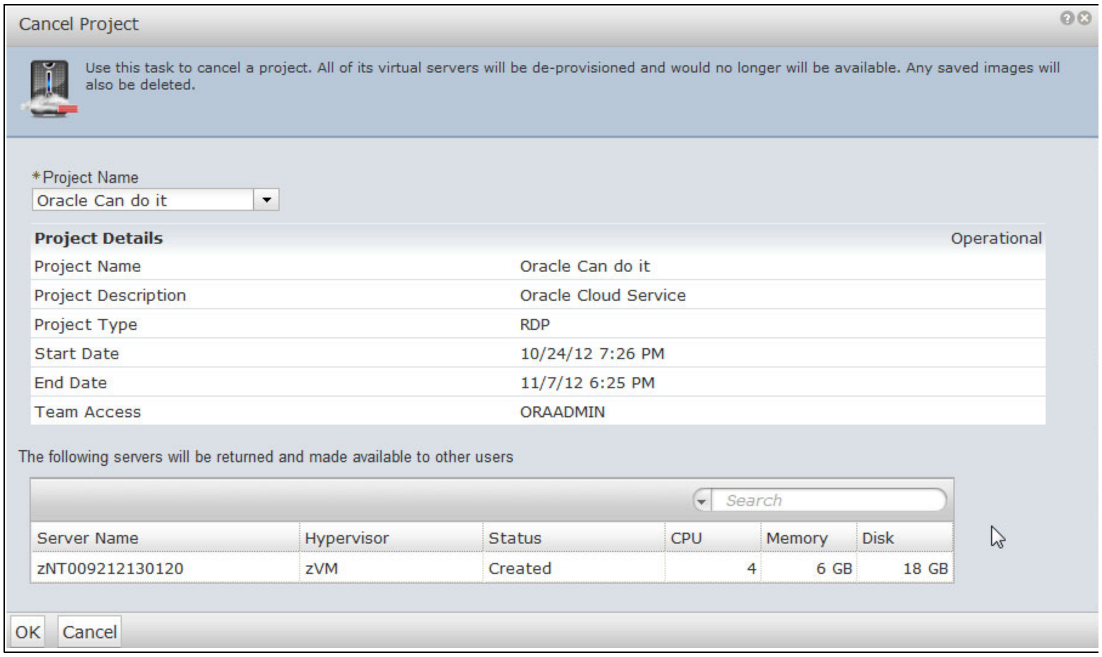


Figure 11-52 Cancel a project

You receive an email from Tivoli Service Automation Manager administrator. Follow the progress the same way as described in “Requesting a new Oracle Service” on page 274.

11.7.7 Unregistering a z/VM project

When you want to delete a server image, complete the following steps from the self-service interface:

1. Click the **Unregister Image** button.
2. Select the image to be unregistered and click **OK**.
3. Confirm the action by clicking **OK** again.
4. Follow the progress in My Requests portlet.

11.8 Summary

For more information, see the following resources:

- *z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 11 SP1*, SG24-7931-00
- *Directory Maintenance Facility Tailoring and Administration Guide; Version 6 Release 1*, SC24-6190-00
- *Deploying a Cloud on IBM System z*, REDP-4711-00
- *Provisioning Linux on IBM System z with Tivoli Service Automation Manager*, REDP-4663-00
- *Installing Oracle 11gR2 RAC on Linux on System z*, REDP-4788
- *Experiences with a Silent Install of Oracle Database 11gR2 RAC on Linux on System z (11.2.0.3)*, REDP-9131
- *Tivoli Service Automation Manager Version 7.2.2: Installation and Administration Guide*, SC34-2657-00
- Tivoli Information Center, which is available at this website:

  http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/index.jsp?topic=%2Fcom.ibm.tsam.doc_7.2%2Ft_config_zvm_setup.htmlhttp://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/index.jsp?topic=%2Fcom.ibm.tsam.doc_7.2%2Ft_config_zvm_setup.html

12

Using z/Pro as a Cloud infrastructure for Oracle

Attention: The following command conventions are used in this chapter:

- z/VM commands are prefixed with ==>
- Linux commands that are running as root are prefixed with #

This chapter describes implementing a zCloud architecture for a development logical partition (LPAR) that supports Oracle on Linux on System z.

By using Cloud architecture, application developers can create their servers as needed with the expiration of servers that are set to a defined date.

This chapter includes the following topics:

- zPRO introduction
- Cloud implementation overview
- Shared Binary Linux implementation: SUSE Linux Enterprise Server 11 SP2
- Shared Binary Oracle implementation: Oracle 11g
- Cloning 100 Oracle Servers for Development: Oracle 11g
- References

12.1 zPRO introduction

The administration and use of a z/VM system can be a highly technical and specialized process. The administrator requires certain skills, most notably of which is knowing how to navigate a 3270 terminal interface. Users that are new to the platform or unfamiliar with the intricacies of z/VM might see that as a deterrent to getting started.

zPro from Velocity Software makes the administration of z/VM significantly easier by providing a web-based front end to most of the tasks that are done regularly. zPro enables cloud computing through which a user can create any number of virtual machines to fit their business requirements. If an external security manager (ESM) is available, such as, the Resource Access Control Facility (RACF) Security Server for z/VM, the web-based interface also extends to it.

Users of zPro can be restricted to perform only certain tasks that are relevant to their responsibilities. Functions, such as, creating virtual machines, allocating disk space, changing memory sizes, administering the Shared File System (SFS), or even RACF can be tailored to the needs of the specific user.

12.2 Cloud implementation overview

This section provides an overview of a Cloud implementation.

12.2.1 Requirements

Managing a large z/VM system is a specialized skill. It involves understanding the concepts and facilities that z/VM provides and how to configure and use them to construct the environment that is needed for your organization. The z/VM Systems Management Application Programming Interface (SMAPI) simplifies the management of z/VM with a standardized, platform-independent programming interface that can reduce the required skills in this environment.

SMAPI must be configured so that it can be used. Because the intent is to use zPro with the SMAPI, these configuration changes include those that are required for zPro.

The following overall steps are required:

1. Directory Maintenance (Dirmaint) changes
2. SMAPI configuration

The zPRO software also is needed for Cloud implementation.

12.2.2 Dirmaint configuration

Because SMAPI works with Dirmaint, changes are needed to the Dirmaint configuration.

Complete the following steps to make the necessary changes:

1. Log on to Dirmaint.
2. Press Enter twice to stop the application.

   The `AUTHFOR CONTROL` file is used to tell Dirmaint, which users (or virtual machines) are authorized to issue commands.

3. Edit the file by using the following command:

   ==> **xedit authfor control e**

   The entries that are shown in Figure 12-1 must be added to the file for the SMAPI and zPro servers.

```
ALL VSMGUARD  *       140A ADGHOPSM
ALL VSMGUARD  *       150A ADGHOPSM
ALL VSMWORK1  *       140A ADGHOPSM
ALL VSMWORK1  *       150A ADGHOPSM
ALL VSMWORK2  *       140A ADGHOPSM
ALL VSMWORK2  *       150A ADGHOPSM
ALL VSMWORK3  *       140A ADGHOPSM
ALL VSMWORK3  *       150A ADGHOPSM
ALL ZPRO01    *       140A ADGHOPSM
ALL ZPRO01    *       150A ADGHOPSM
ALL ZPRO02    *       140A ADGHOPSM
ALL ZPRO02    *       150A ADGHOPSM
ALL ZPRO03    *       140A ADGHOPSM
ALL ZPRO03    *       150A ADGHOPSM
```

Figure 12-1 AUTHFOR CONTROL file

4. Create a file that is called `CONFIGSM DATADVH D` (as shown on Figure 12-2) by using the following command:

   ==> **xedit configsm datadvh d**

```
/* SMAPI config */
   ALLOW_ASUSER_NOPASS_FROM=           VSMGUARD *
   ALLOW_ASUSER_NOPASS_FROM=           VSMWORK1 *
   ALLOW_ASUSER_NOPASS_FROM=           VSMWORK2 *
   ALLOW_ASUSER_NOPASS_FROM=           VSMWORK3
   ALLOW_ASUSER_NOPASS_FROM=           ZPRO01   *
   ALLOW_ASUSER_NOPASS_FROM=           ZPRO02   *
   ALLOW_ASUSER_NOPASS_FROM=           ZPRO03   *
   ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.TCP= DVHXNE EXEC
   ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.UDP= DVHXNE EXEC
```

Figure 12-2 CONFIGSM DATADVH file

   The entries **ALLOW_ASUSER_NOPASS_FROM** permits the SMAPI server machines and zPRO server machines to issue Dirmaint calls as another user.

   The entries **ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT** notifies the SMAPI of changes that are made to the directory.

5. Restart Dirmaint by using the following command:

   ==> **dvhbegin**

12.2.3 Dirmaint direct access storage device configuration

zPro relies on Dirmaint's view of the disk subsystem to allocate minidisks. Make sure that the disk volumes that are needed for minidisks are available to Dirmaint and, optionally, that they are placed into groups (disk pools) for more generic placement of minidisks.

Defining disk volumes to Dirmaint and placing those volumes into groups is done in the `EXTENT CONTROL` file. This file is under the control of Dirmaint. To see the contents of the file, run the following command:

```
==> dirm send extent control
```

The `EXTENT CONTROL` file is transferred from Dirmaint to the reader queue of the virtual machine from where the command is issued.

Modify the `REGIONS` section by adding disk volumes for minidisk allocation. The start and end fields indicate the part of the volume that is to be used for allocation and are typically `START` and `END` so that the entire volume is used. Pay special attention to the device type because it should be the exact device type of the volume to make best use of the space and prevent allocation errors.

The `GROUPS` section also must be changed by adding regions (volumes) to existing groups, or creating groups from other volumes in the `REGIONS` section.

When modifications are complete, send the file back to Dirmaint by the following command:

```
==> dirm file extent control
```

Then, tell Dirmaint that the file was updated by using the following command:

```
==> dirm rldextn
```

The new `REGIONS` and `GROUPS` definitions are available for use by Dirmaint and zPro.

12.2.4 SMAPI implementation

If your system is not in an ensemble, the following changes for SMAPI must be made on `MAINT` to turn off ensembles for SMAPI:

- Access the 193 disk, as shown in the following example:

```
==> acc 193 f
Ready; T=0.01/0.01 15:20:22
```

- Access the SMAPI worker server data directory in read/write mode, as shown in the following example:

```
==> acc vmsys:vsmwork1.data g (forcerw
Ready; T=0.01/0.01 15:20:35
```

- Copy `DMSSISVR NAMES` from the 193 disk to the directory, as shown in the following example:

```
==> copy dmssisvr names f = = g
Ready; T=0.01/0.01 15:20:50
```

- Xedit the file on the directory, as shown in the following example:

```
==> x dmssisvr names g
Ready; T=0.01/0.01 15:25:15
```

- Comment out the section of the file at the end under the comment, as shown in Figure 12-3.

```
************************************************************
*** the following machines are only available in ensembles ***
************************************************************

* Default Management Network Server
*:server.VSMREQIM
*:type.REQUEST
*:protocol.AF_MGMT
*:address.INADDR_ANY
*:port.44446

* Primary Vswitch Controller
*:server.DTCENS1
*:type.VCTRL

* Backup Vswitch Controller
*:server.DTCENS2
*:type.VCTRL

* Management Guest
*:server.ZVMLXAPP
*:type.MG
```

Figure 12-3 DMSSIVR NAMES

When the configuration steps are completed, start the SMAPI server machines by using the following command from an authorized user:

```
==> XAUTOLOG VSMGUARD
```

This virtual machine automatically starts the other SMAPI server machines. To verify that the SMAPI is running and listening to requests, run the following commands:

```
==> vmlink tcpmaint 592
==> netstat
```

As shown in Figure 12-4, the results show a listener at port 44444 and 55555, which are SMAPI TCP/IP services.

```
VM TCP/IP Netstat Level 620       TCP/IP Server Name: TCPIP

Active IPv4 Transmission Blocks:

User Id  Conn    Local Socket            Foreign Socket          State
---- --  ----    ----- ------            ------- ------          -----
FTPSERVE 1001    *..FTP-C                *..*                    Listen
INTCLIEN 1079    *..TELNET               *..*                    Listen
INTCLIEN 1073    9.12.4.200..TELNET      9.12.5.239..2037        Established
INTCLIEN 1044    9.12.4.200..TELNET      9.12.5.139..4862        Established
INTCLIEN 1083    9.12.4.200..TELNET      9.12.5.239..2800        Established
INTCLIEN 1056    9.12.4.200..TELNET      9.57.138.250..2986      Established
ZTCP     UDP     *..1024                 *..*                     UDP
ZADMIN   1003    *..81                   *..*                    Listen
ZWEB02   1055    *..80                   *..*                    Listen
ZWEB04   1031    *..80                   *..*                    Listen
ZWEB01   1063    *..80                   *..*                    Listen
ZWEB05   1074    *..80                   *..*                    Listen
ZWEB03   1072    *..80                   *..*                    Listen
VSMREQIN 1036    *..44444                *..*                    Listen
VSMEVSRV 1067    *..55555                *..*                    Listen
```

Figure 12-4 NETSTAT output that validates SMAPI

12.2.5 zPRO implementation

The installation of the zPro product is done by using the Velocity Software product management virtual machine ZVPS. For more information about the zPRO installation process, see the Velocity Software Installation Guide.

After the installation is complete, zPro should be configured for the environment. Configuration is done within the Velocity Software maintenance environment (VSIMAINT) by clicking **Edit Run Time Configuration Files**. Move to that line and press F2. A list of configuration files for that product is shown. Move the cursor to CONFIG ZPRO and press F2. The file is shown, as shown in Figure 12-5 on page 285.

```
ZPROCFG                      Velocity Software Inc.           ZPRO PROD1320
                            CONFIG ZPRO Configuration

First server                          ZPRO01
Server list   ZPRO01 ZPRO02 ZPRO03
Log server                            ZPROLOG
zPRO Log Location            SFSZVPS:ZPROLOG.LOGS
zPRO Log Prune Age                    30
SMAPI-authorized userid               ZPRO01
SMAPI-auth userid password            VSIZPRO
Show passwords?                       NO
Socket timeout                        60
TCP/IP userid                         TCPIP
z/VM image hostname                   ZVM
SMAPI Port                            44444
Client Port                           8889
Network domain               ITSO.COM
Directory manager                     DIRMAINT
Dir Mgr Command Location     MAINT 19E
ESM                                   NONE
DIRM2RACF                             NO
Directory Mgr Src Location   DIRMAINT 1DF
Directory Mgr Cfg Location   DIRMAINT 11F
RACF default owner                    IBMUSER
RACF default group                    SYS1
Proxy user for batch RACF processing  ZPROXY
Proxy user to set unexpired passwords ZPROXY
RSCSAUTH config location     RSCSAUTH 191
Max logon time                        480
E-mail server                VM:SMTP
E-mail from address          ZPRO@ITSO.COM
Expire active servers                 SHUTDOWN
IP parameter files location  ZPRO 391
IP parm minidisk read pw              READ
IP parm minidisk write pw             WRITE
Debug                                 YES
Javascript debugging                  NO
zPRO Hourly Exits



PF1: Help    PF2: Validate/Save   PF3: Exit
                                             PF10: Default   PF12: Cancel
```

Figure 12-5 Configuring zPRO for zCLOUD

In this window, there are various zPro configuration parameters. Most of these can be left to default. Verify that the settings are correct if RACF is in use and modify Network Domain and the E-mail from address sections for your organization. Make sure that the SMAPI-auth user ID password is the CP (or RACF) password for the `ZPRO01` virtual machine.

When the changes are complete, press F2 to validate the changes, then press F2 again to save them.

Now that the configuration is complete, zPro can be started. Use the **XAUTOLOG** command to start ZPRO01, as shown in the following example:

```
==> xautolog zpro01
```

This automatically brings up the other zPro servers.

12.2.6 Using zPro

To start using zPro, open a web browser and enter the following URL:

http://<VM-IP-Address>/zpro

A window similar to the one that is shown in Figure 12-6 opens.

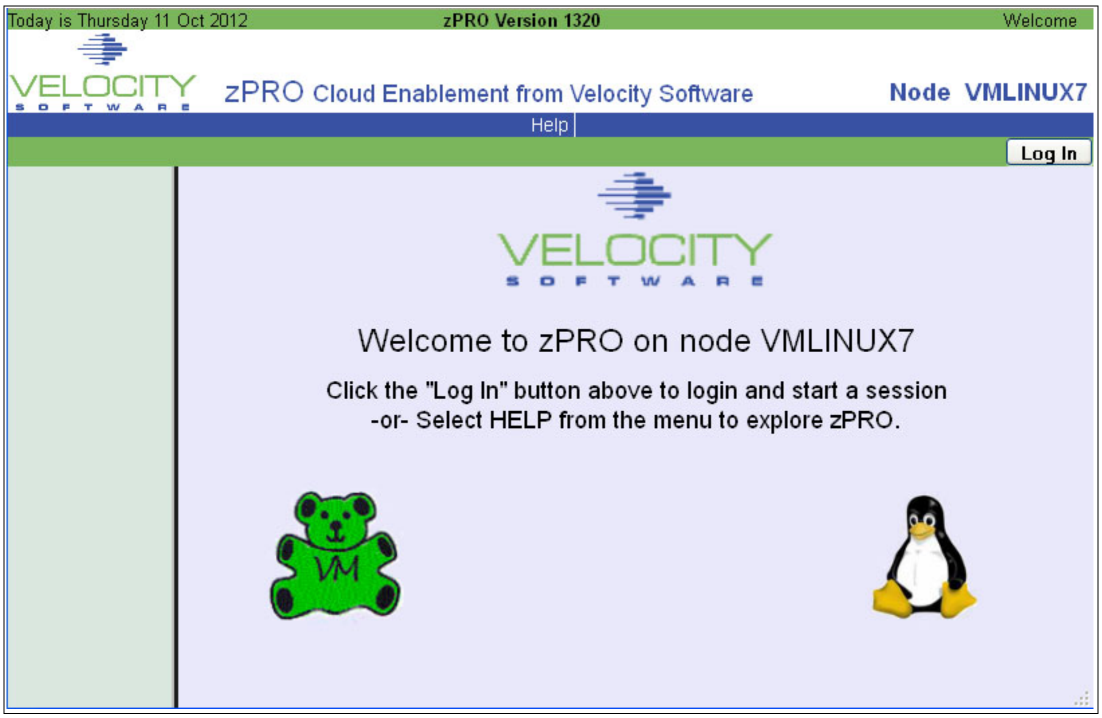


Figure 12-6 zPRO home page

On the right side of the page under the node name is a Log In button. Click the button and the zPRO Login window opens, as shown in Figure 12-7.

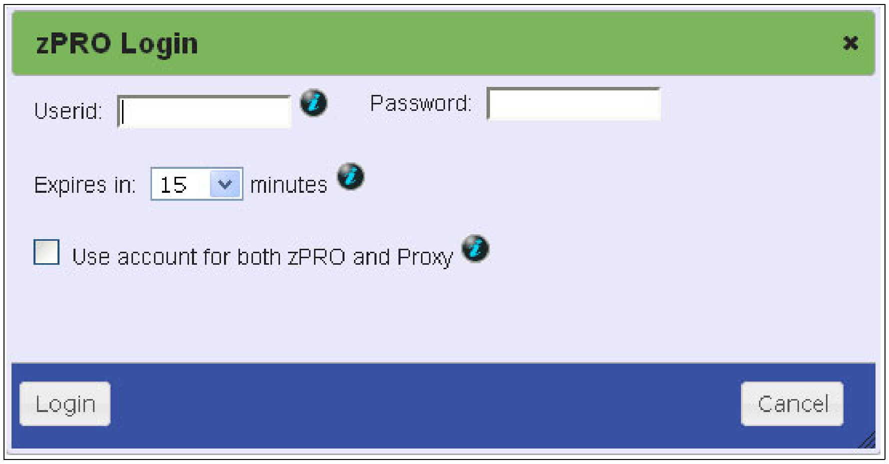


Figure 12-7 zPRO Login window

The following information is entered or set in this window:

- Userid

  By default, zPro includes a User ID of `ZPRO`, which is a full administrator and as such can perform any zPro related function.

- Password

  The default password of the `ZPRO` user is CP or RACF.

- Expires In

  This is the amount of time that the zPro session should be active. After the elapsed time, the session is automatically logged off. The maximum time is 480 minutes (8 hours).

  For auditing and control purposes, zPro uses the concept of a proxy to indicate which virtual machine is used to issue RACF and SFS commands. Without the proxy, the web servers issue those commands. In most instances, it is unlikely that security allows the web servers to have the amount of authority that is required for the proper use of zPro. The proxy is a virtual machine that has the correct amount of authority to issue the RACF and SFS commands and also satisfy the auditing requirement to assign responsibility for the commands issued.

- Use account to both zPro and Proxy

  This is used to tell zPro that the user that is logging on is also the proxy. The virtual machine that is represented by that user ID is XAUTOLOGged for each set of RACF or SFS commands that are needed during a zPro session. If this option is not selected, another authentication window opens after the Login button is clicked to allow the user to enter the user ID and password of the proxy machine that has the correct amount of authority to issue the RACF and SFS commands.

When the login process is complete, the window that is shown in Figure 12-8 opens.

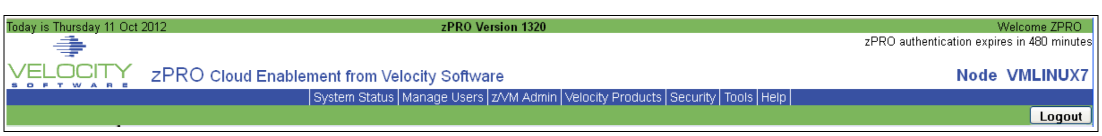


Figure 12-8 zPRO Portal

The user ID that was logged in is shown in the upper right. The timer underneath ticks down as an indication of how much time is left in the session. The VM node name is under the timer, and the Logout button is under the node name. The zPro release number is in the upper middle of the window.

The blue bar is the main menu window. Hovering the mouse over a menu shows the contents. Click a menu item to select that function. Figure 12-9 shows the Manage Users menu.

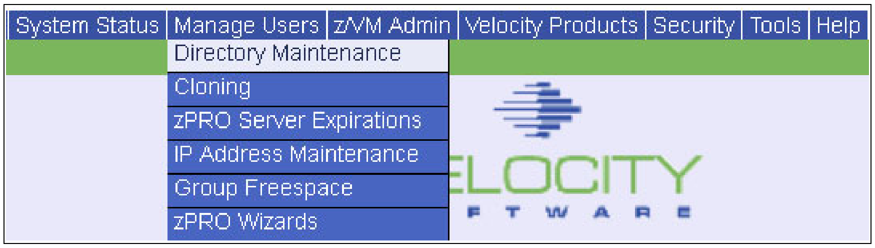


Figure 12-9 Manage Users menu

For example, click **Directory Maintenance** to show all of the defined virtual machines. Figure 12-10 on page 289 shows the resulting Directory Maintenance window with the defined virtual machines listed on the left, Factories along the top and the Work zone underneath the factories. A Factory is a function. Drag a virtual machine to a factory to perform that function on it. At the bottom of the window is the Action Log, which is used to display messages about functions that are performed.

In the User List on the left side of the window, some virtual machines are displayed in different colors. Those that have a purple background are logged on. Virtual machines that have a gold background are considered Gold machines, or template machines. They can be used for cloning new virtual machines.

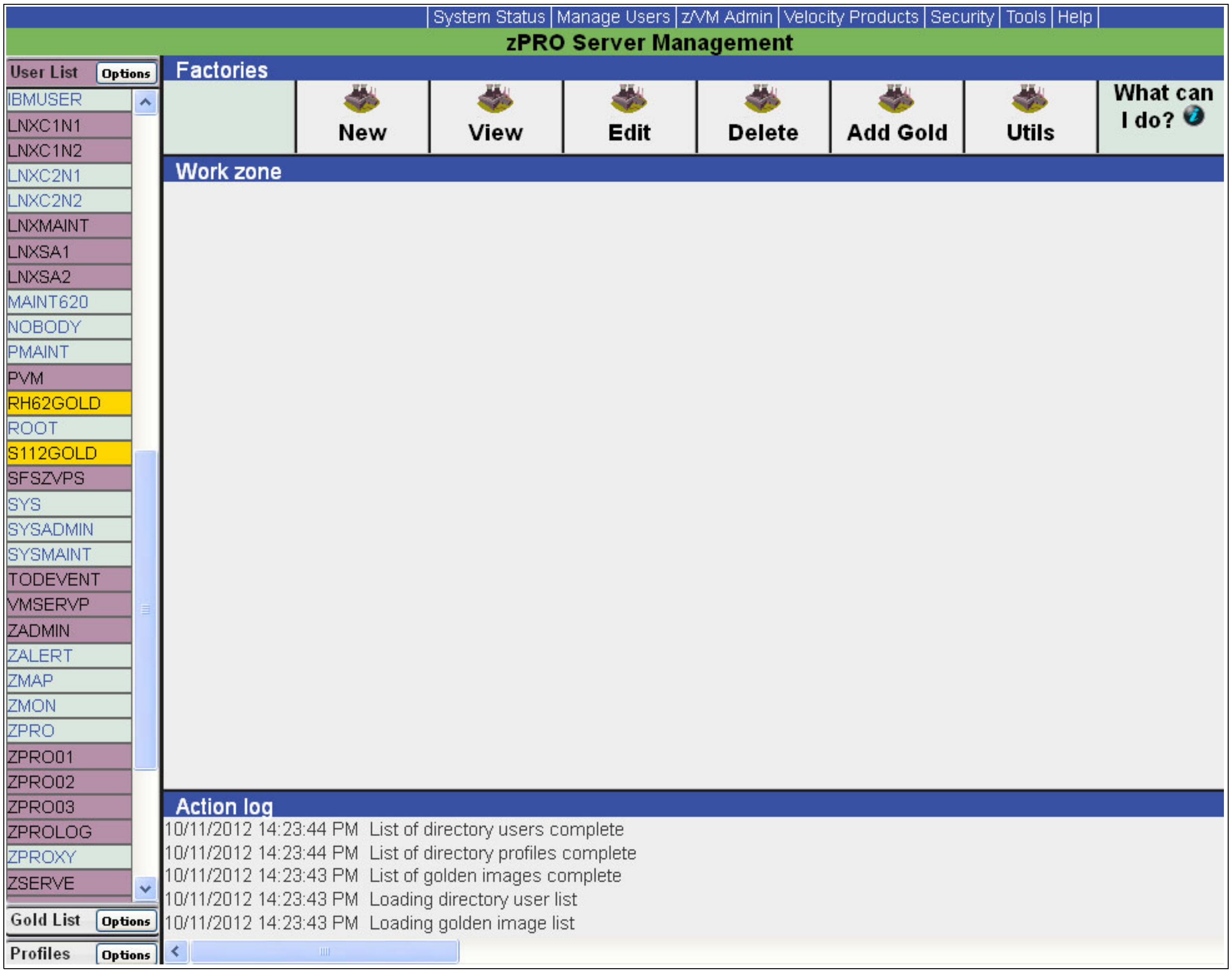


Figure 12-10 Directory Maintenance window

IP address set up

zPro can assign IP addresses to virtual machines that are cloned. The range of IP address that zPro should assign are entered in the IP Address Maintenance window under the Manage User's menu, as shown in Figure 12-11.

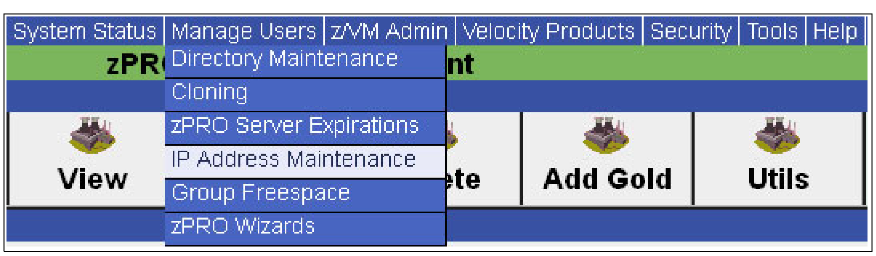


Figure 12-11 Selecting IP Address Maintenance

Figure 12-12 on page 290 shows the maintenance window. Enter the start and end of the IP address range in the fields that are provided and click **Add**. zPro display again the window with the entire range in the Available IP addresses section. Addresses that cannot be used for any reason can be removed by entering the address (or range) and clicking **Delete** on the far right. They are removed from the list of available addresses.

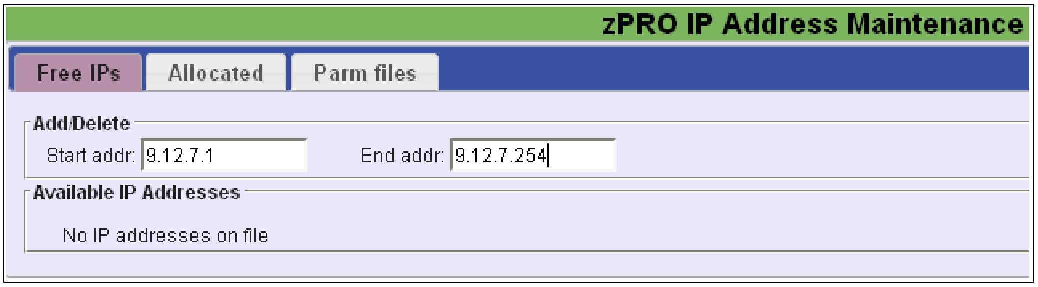


Figure 12-12 IP Address Maintenance

IP Address Assignment

Changing the IP address of any clones that are created by zPro requires action on the Linux machine as it comes up for the first time. A shell script is included as part of zPro to accomplish this task.

When the IP addresses are added to zPro, they are put on a list of available IP addresses. During the cloning process, an IP address from the available list is selected. It is assigned to the clone by writing it to a file on a common minidisk with the name of the new virtual machine. The new hostname and interface device address are also written to the file. If there are multiple cloned machines, they are assigned IP addresses, starting with the first one that is selected from the list.

The golden image requires the shell script to be installed into `/etc/rc.d/boot.d`. The **chkconfig** command is used to turn on the new boot service. The following commands demonstrate the process:

```
# cp boot.vsisetup /etc/rc.d/boot.d
# cd /etc/rc.d/boot.d
# chkconfig boot.vsisetup on
```

Verify that it is turned on, as shown in the following example:

```
s112gold:/etc/rc.d # chkconfig -A --list | grep boot.vsisetup
boot.vsisetup   0:off  1:off  2:off  3:off  4:off  5:off  6:off  B:on
```

When the Linux machine is first booted after cloning, the script runs and then reads the file that was created for it on the common minidisk. The new IP address and hostname are obtained from the file and the appropriate files are changed on Linux. This process occurs before the network starts. After the script ends, the boot process continues normally. When networking is started, the new IP address and hostname are set and there is more re-boots are not needed.

12.2.7 Cloning

Cloning is the process of creating a virtual machine that looks exactly like an existing virtual machine and copying any read/write disk areas from the source machine (the gold image) to new locations for the destination virtual machine (the clone). The intent is to mass-produce new virtual machines from an existing virtual machine.

In addition to the system administration capabilities, zPro can create virtual machine clones. To clone a machine, it must be in the gold list. To add a virtual machine to the gold list, go to the Directory Maintenance page, drag the name of a machine from the User List on the left of the display to the Add Gold factory. Messages in the Action Log on the bottom confirm when the process is completed.

Linux should be installed and prepared to start when the clone is initially loaded. It should contain the software packages and configuration settings that are generally required for any Linux machine in your organization. Before it is cloned, the gold image Linux must be logged off. As protection against this, zPro does not allow the cloning of an active gold image.

Any number of gold images can be created. They can be plain Linux machines, or a Linux machine setup for a specific task, such as, Oracle or IBM WebSphere.

To start the cloning process, select **Cloning** under the Manage Users menu. The zPRO Cloning window opens, as shown in Figure 12-13.

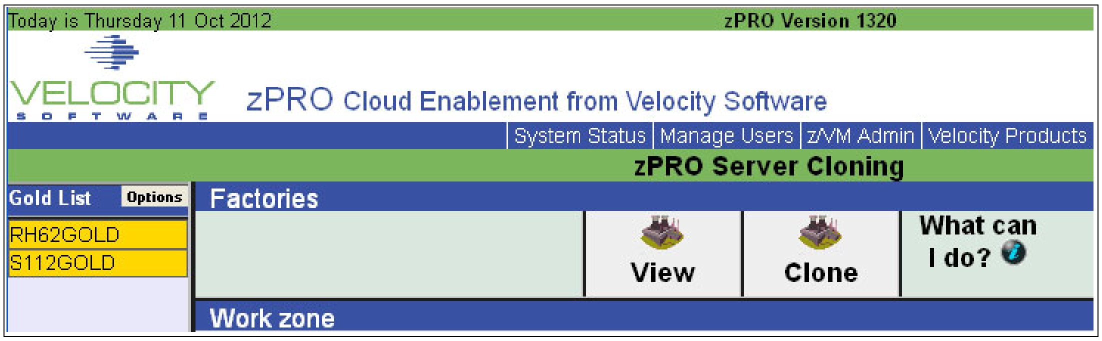


Figure 12-13 zPro Cloning Screen

Defining the cloud administrator

zPro has a built-in security infrastructure that allows an administrator to control who has access to the environment and what they can see or modify.

A default administrator is delivered called `ZPRO`. A virtual machine definition must also exist for any new users that are defined to zPRO. The virtual machine definition for a zPRO user does not need to have any specific CP privileges.

To create a new zPro user, select **zPRO authorizations** from the Security menu, as shown in Figure 12-14.

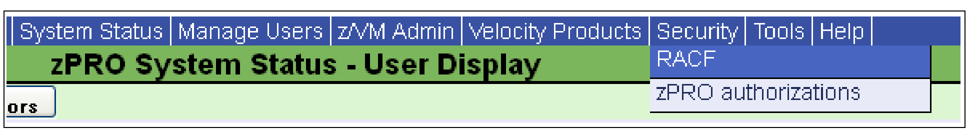


Figure 12-14 zPro Security menu

Figure 12-15 on page 292 shows the zPro User Authorization window. The User List on the left side of the window shows the defined zPro users. The Factories are tasks that can be run. By using the New Factory, you can create a zPro user from scratch or based on a user. By using the Edit Factory, you can modify a zPro user. By using the Delete Factory, you can delete a zPro user.

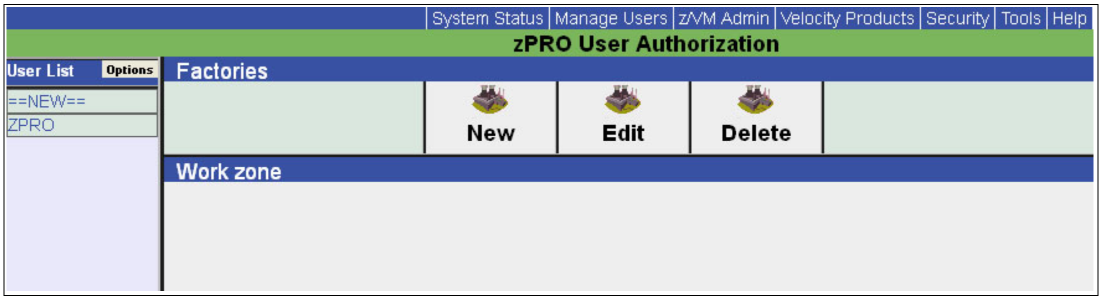


Figure 12-15 zPro User Authorization

An existing CMS user can become a zPro user by creating a zPro profile for them. Dragging the New user to the New Factory displays a blank user profile that can be used to build a zPro user, as shown in Figure 12-16.

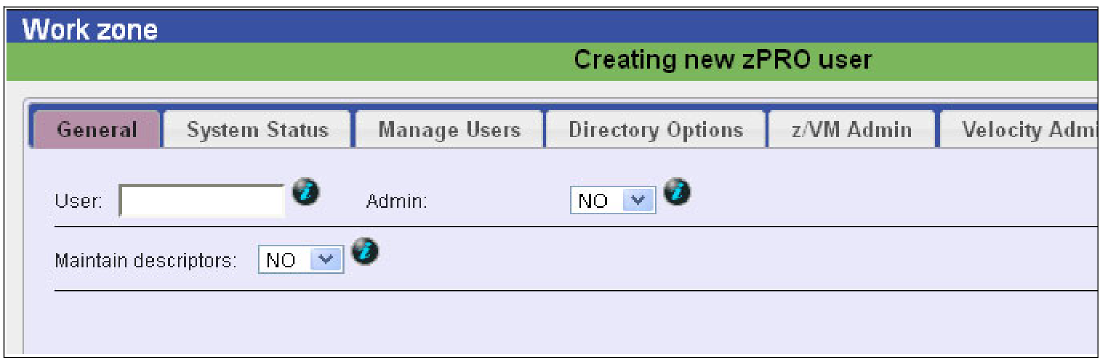


Figure 12-16 Creating a zPro user

In the Work Zone, the tabs across the top are the various security areas that might need to be completed, based on the level of access the new user requires. Start by completing the User field and selecting whether this new user is an Admin. Then, select the tabs one at a time and complete the fields in those tabs to authorize the user for various functions.

Any field that has the More Information graphic  is a link to help for the specific field.

Clicking the More Information graphic for the Admin drop-down menu shows a window in which the selections are described and how they affect the user, as shown in Figure 12-17. Click the **Return** button to close the window.

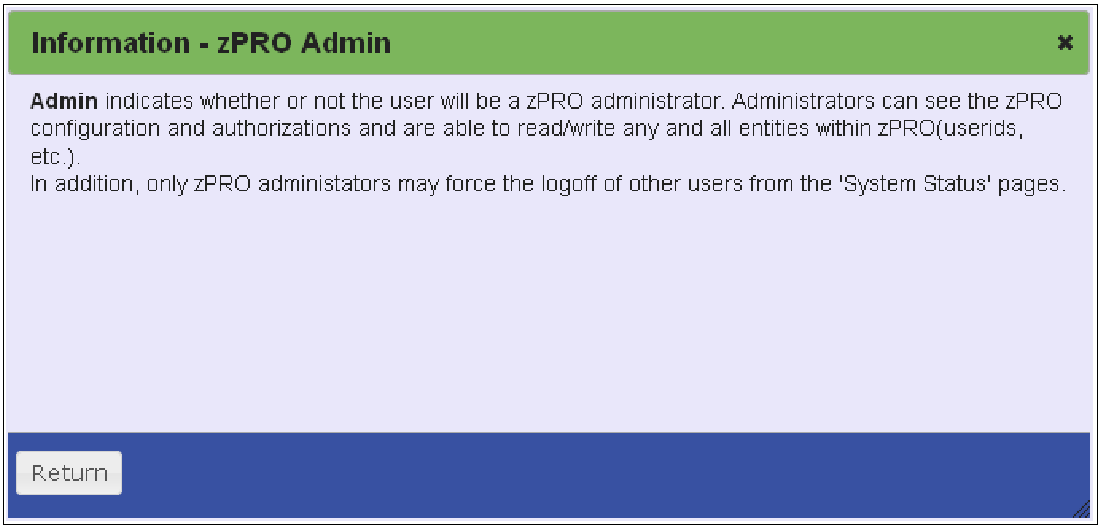


Figure 12-17 An example of more information for a zPro field

Another way to create a zPro user is to drag an existing user to the New Factory. This creates a zPro user profile that has the same authorizations as the old user. It is necessary to complete the User field to give the new user a name.

Figure 12-18 shows a new user that is called `JADMIN`. It is based on the delivered ZPRO user.

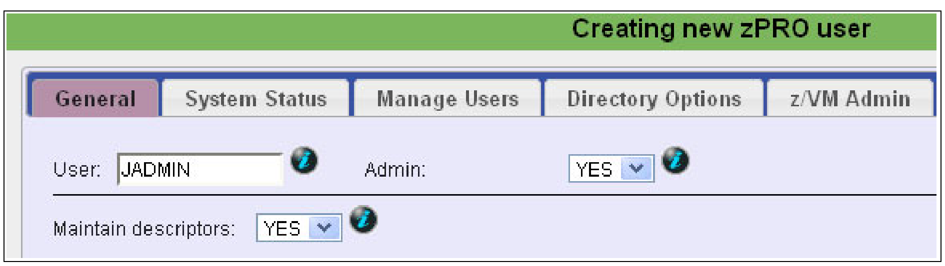


Figure 12-18 Creating a zPro user from an existing user profile

When all of the fields in the various tabs are complete (or modified), click **Save** to save the new profile.

Any new users that are defined to zPro also must be authorized for Dirmaint. Add them to `AUTHFOR CONTROL`, as shown in the following example:

```
ALL JADMIN * 140A ADGHOPSM
ALL JADMIN * 150A ADGHOPSM
```

12.3 Shared Binary Linux implementation: SUSE Linux Enterprise Server 11 SP2

SUSE Linux Enterprise Server 11 SP2 provides a well-documented, read-only root implementation. It is used during our testing to conserve disk space and have a consistent root file system for massive cloning operations.

Read-only root was demonstrated previously, but it always presented many challenges, including setup and maintenance. A Redpaper™ *Sharing and maintaining Linux under z/VM* was written that documented and demonstrated the process. The implementation of shared read-only root is much cleaner on SUSE Linux Enterprise Server 11 SP2 and is a supported environment.

12.3.1 Architecture

Two virtual machines were used as templates for read-only root. The first machine, `S11ROMST`, contains the read/write root file system for the remaining images. All root file system changes to make read-only root work properly was done there. It is a complete and functioning Linux virtual machine. The second machine, `S11ROGLD`, was created as an image of the first, including its root file system. Then, the directory entry was changed. Replacing the root file system minidisk with a link to the root file system of `S11ROMST`. Then, `S11ROGLD` can be cloned.

Figure 12-19 shows the directory entry view function of zPro displaying `S11ROMST`. It occupies a 3390 model 9 DASD. There is also a link to the `LNXSA3` 101 minidisk that contains the Oracle binaries. As each machine is cloned, Oracle can be installed.

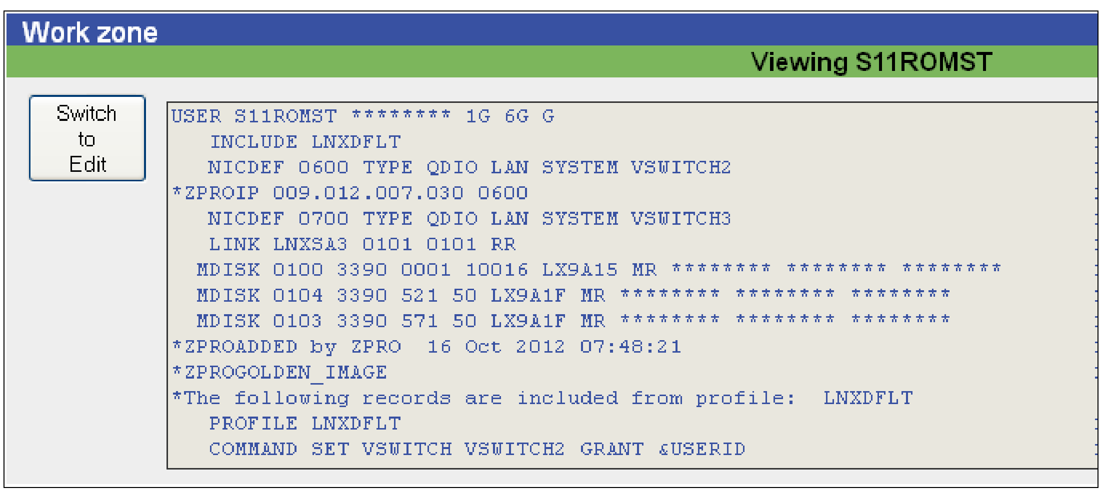


Figure 12-19 S11ROMST definition in User Directory

A private subnet was used for the cloned machines that allowed the creation of more virtual machines with no concern about addressing limitations on the primary network. The `LNXSA3` machine was used as a router between the primary network and the new subnet.

When the networking parameters were established, `LNXSA3` was set up to properly route traffic to the new network. This includes turning on Enable IP Forwarding on the Routing tab in YaST. Additionally, the following command was issued on any machine that needed access to the new clones:

```
# route add 10.1.1.0 mask 255.255.255.0 9.12.7.5
```

The Windows **route print** command on our testing machine then displayed the new route that was created:

```
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0         9.12.4.1      9.12.5.239      10
         9.12.0.0    255.255.240.0       9.12.5.239      9.12.5.239      10
       9.12.5.239  255.255.255.255        127.0.0.1       127.0.0.1      10
    9.255.255.255  255.255.255.255       9.12.5.239      9.12.5.239      10
         10.1.1.0    255.255.255.0         9.12.7.5      9.12.5.239       1
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
        224.0.0.0        240.0.0.0       9.12.5.239      9.12.5.239      10
  255.255.255.255  255.255.255.255       9.12.5.239      9.12.5.239       1
Default Gateway:          9.12.4.1
```

The `S11ROMST` virtual machine and its clones use the new subnet. In YaST, configure it with an IP address on the 0700 network device (`eth1`). After it is functioning, the 0600 network device (`eth0`) can be removed and deleted from the directory entry.

zPro must also know about the new subnet. From Manage Users, click **IP Address Maintenance**. Add the IP addresses for the new subnet, as shown in Figure 12-20.

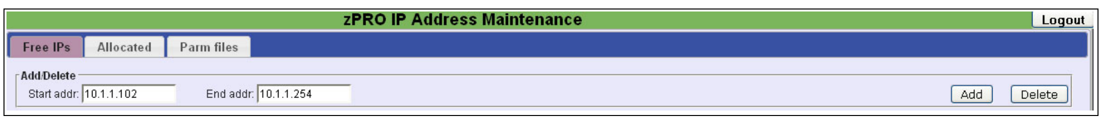


Figure 12-20   IP Address range for the new subnet

12.3.2 Adding kernel modules

To allow for the use of cooperative memory management and issuing CP commands from Linux, two kernel modules were added to the Linux startup. The modules are called `cmm` and `vmcp`. In YaST, browse to **System** → **Kernel** → **MODULES_LOADED_ON_BOOT**. Enter their names, as shown in Figure 12-21.

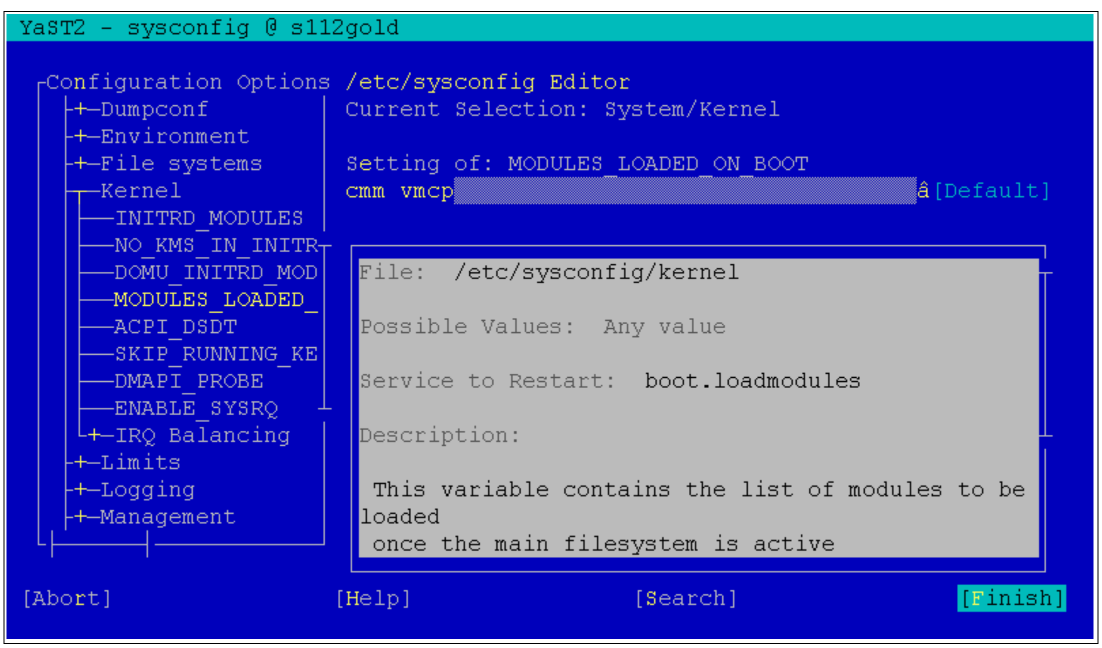


Figure 12-21   Additional kernel modules

12.3.3 Setting up for read-only root

There are a few steps to set up the read-only root environment on SUSE Linux Enterprise Server 11 SP2.

Modify /etc/rwtab

A new file called `/etc/rwtab` must be modified. It contains a list of the files or directories that should be available in read/write mode. Our `/etc/rwtab` file is shown in the following example:

```
dirs    /var/cache/man
#dirs   /var/gdm
dirs    /var/lock
dirs    /var/log
dirs    /var/run

empty   /tmp
#empty  /var/cache/foomatic
#empty  /var/cache/logwatch
#empty  /var/cache/mod_ssl
#empty  /var/cache/mod_proxy
#empty  /var/cache/php-pear
empty   /var/cache/systemtap
#empty  /var/db/nscd
#empty  /var/lib/dav
#empty  /var/lib/dhcp
#empty  /var/lib/dhclient
empty   /var/lib/dhcpcd
#empty  /var/lib/php
#empty  /var/lib/ups
empty   /var/tmp
empty   /var/tux
empty   /media

files   /etc/adjtime
files   /etc/ntp.conf
files   /etc/resolv.conf
files   /etc/yp.conf
files   /etc/lvm/.cache
files   /var/account
files   /var/adm/netconfig/md5
files   /var/arpwatch
files   /var/cache/alchemist
files   /var/lib/iscsi
files   /var/lib/logrotate.status
files   /var/lib/ntp
files   /var/lib/xen
files   /root
files   /var/lib/gdm
#files  /etc/X11/xdm

state   /var/lib/misc/random-seed
state   /etc/ssh
state   /etc/fstab
state   /etc/HOSTNAME
state   /etc/hosts
state   /etc/sysconfig/network
```

The last four lines of the file contain the entries that were added to support the process of modifying IP addresses dynamically. The file is only referenced when read-only root is used.

Creating state and scratch minidisks

Two small minidisks must be added to the Linux machine. They are used to hold the read/write files that are listed in `/etc/rwtab`. The files that are listed there are bind-mounted onto the two new disks.

The size of these minidisks is based on the size of the files that are listed in `/etc/rwtab`. The files that are listed in the State section are mapped to the state disk; the rest of the files are mapped to the scratch disk. For our template read-only image, two 50-cylinder minidisks were used.

Modifying /etc/zipl.conf

The file `zipl.conf` must be modified to tell the boot procedure that read-only root is used and to identify the state and scratch disk devices. The following example shows the `SLES11_SP2` boot section from our read-only root template:

```
[SLES11_SP2]
    image = /boot/image-3.0.13-0.27-default
    target = /boot/zipl
    ramdisk = /boot/initrd-3.0.13-0.27-default,0x2000000
    parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 hvc_iucv=8
TERM=dumb vmpoff=LOGOFF vmhalt=LOGOFF"
```

The following example shows the modified boot section. The parameters keyword were changed to add `readonlyroot`, in addition to the scratch and state subkeywords. The values of scratch and state are the new minidisks that were added to the Linux machine to support the read/write parts of the file system:

```
[SLES11_SP2]
    image = /boot/image-3.0.13-0.27-default
    target = /boot/zipl
    ramdisk = /boot/initrd-3.0.13-0.27-default,0x2000000
    parameters = "readonlyroot scratch=/dev/disk/by-path/ccw-0.0.0103-part1
state=/dev/disk/by-path/ccw-0.0.0104-part1
root=/dev/disk/by-path/ccw-0.0.0100-part1 hvc_iucv=8 TERM=dumb vmpoff=LOGOFF
vmhalt=LOGOFF"
```

Symlink /etc/mtab

The next step is to create a symlink to replace `/etc/mtab` with `/proc/mounts`. This is an important step that allows modification of the mount table because the `/etc` directory is read-only. Enter the following command:

```
s11romst:/etc # ln -sf /proc/mounts /etc/mtab
s11romst:/etc # ll mtab
lrwxrwxrwx 1 root root 12 Oct 16 15:11 mtab -> /proc/mounts
```

Modify /etc/fstab

At boot time, it is normal for Linux to check the integrity of any file systems that it uses. This check is done at regular intervals. Because a read-only file system cannot be checked, the file `/etc/fstab` must be modified to indicate that the root file system should not be checked. The following example shows the `/etc/fstab` file before it is changed to turn off file system checking:

```
/dev/system/root     /                    ext3      acl,user_xattr       1 1
/dev/disk/by-path/ccw-0.0.0200-part1 /boot ext3      acl,user_xattr       1 0
```

The last field on each line indicates the sequence in which a file system is checked. Changing this value to zero for the root file system disables checking.

Running zipl

Run the **zipl** command to create the boot data since `/etc/zipl.conf` was changed.

Reboot

`S11ROMST` was shut down, logged off, and logged back on. When it came back up, the read-only root startup took effect.

The **lsdasd** command shows the disk configuration, including the new disk devices for read-only root, a shown in the following example:

```
s11rogld:/etc # lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   7042MB    1802880
0.0.0101   active      dasdb     94:4    ECKD  4096   21128MB   5409000
0.0.0103   active      dasdc     94:8    ECKD  4096   35MB      9000
0.0.0300   active      dasdd     94:12   FBA   512    256MB     524288
0.0.0301   active      dasde     94:16   FBA   512    512MB     1048576
0.0.0104   active      dasdf     94:20   ECKD  4096   35MB      9000
```

All of the files or directories after the `/var` mount point are mounted over the two new disk devices, as shown in the following example:

```
s11rogld:/etc # df
Filesystem                    1K-blocks    Used Available Use% Mounted on
rootfs                           507624  156020    325404  33% /
udev                             510208     200    510008   1% /dev
tmpfs                            510208       0    510208   0% /dev/shm
/dev/dasda1                      507624  156020    325404  33% /
/dev/mapper/system_vg-opt_lv     253920   16548    224268   7% /opt
/dev/mapper/system_vg-tmp_lv      34764    5056     27916  16% /tmp
/dev/mapper/system_vg-usr_lv    3096336 2158236    780816  74% /usr
/dev/mapper/system_vg-var_lv     516040  120972    368856  25% /var
/dev/dasdf1                       34764     640     32332   2%
/var/lib/readonlyroot/state
/dev/dasdc1                       34764    5056     27916  16%
/var/lib/readonlyroot/scratch
/dev/dasdc1                       34764    5056     27916  16% /var/cache/man
/dev/dasdc1                       34764    5056     27916  16% /var/lock
/dev/dasdc1                       34764    5056     27916  16% /var/log
/dev/dasdc1                       34764    5056     27916  16% /var/run
/dev/dasdc1                       34764    5056     27916  16% /tmp
/dev/dasdc1                       34764    5056     27916  16% /var/lib/dhcpcd
/dev/dasdc1                       34764    5056     27916  16% /var/tmp
/dev/dasdc1                       34764    5056     27916  16% /media
/dev/dasdc1                       34764    5056     27916  16% /etc/ntp.conf
/dev/dasdc1                       34764    5056     27916  16% /etc/resolv.conf
/dev/dasdc1                       34764    5056     27916  16% /etc/yp.conf
/dev/dasdc1                       34764    5056     27916  16% /etc/lvm/.cache
/dev/dasdc1                       34764    5056     27916  16%
/var/adm/netconfig/md5
/dev/dasdc1                       34764    5056     27916  16%
/var/lib/logrotate.status
/dev/dasdc1                       34764    5056     27916  16% /var/lib/ntp
/dev/dasdc1                       34764    5056     27916  16% /root
```

```
/dev/dasdf1                      34764     640     32332    2%
/var/lib/misc/random-seed
/dev/dasdf1                      34764     640     32332    2% /etc/ssh
/dev/dasdf1                      34764     640     32332    2% /etc/fstab
/dev/dasdf1                      34764     640     32332    2% /etc/HOSTNAME
/dev/dasdf1                      34764     640     32332    2% /etc/hosts
/dev/dasdf1                      34764     640     32332    2%
/etc/sysconfig/network
```

12.3.4 Creating a Base Linux golden image

After `S11ROMST` is set up properly as read-only root, a copy (clone) must be made of it to reference a common root file system. This new machine is called `S11ROGLD` and has a link to the `S11ROMST` root file system and its own scratch and state minidisks. Cloning the new machine gives unique Linux images in a small amount of read/write disk space.

Complete the following steps to change `S11ROMST` so that it can be cloned: f

1. From the zPro main menu, move the mouse over Manage Users, then click **Directory Maintenance**. The user list is shown. To filter down the list, move the mouse over the Options button at the top of the list. A small menu appears, as shown in Figure 12-22. Click **Set Filter**.

Figure 12-22 Filtering the user list: Part 1

2. The Create a list filter window opens. Enter `S11*` in the first field, as shown in Figure 12-23. Click **Apply**.

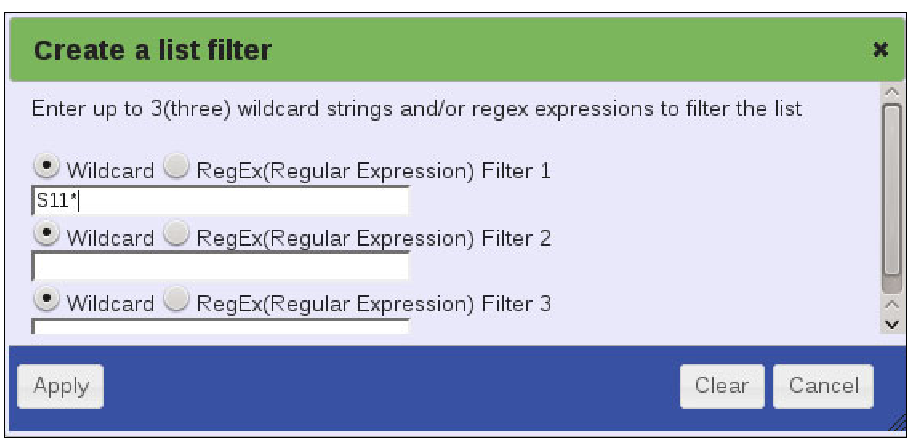


Figure 12-23 Filtering the user list: Part 2

3. The list of users now contain only those that match the filter criteria that was entered. From the filtered list, drag `S11ROMST` to the Add Gold Factory. It is added to the gold list and now it can be cloned.

4. From Manage Users, click **Cloning** to open the window that is shown in Figure 12-24. This shows the cloning window with our gold images on the left of the display, including `S11ROMST`. Drag it to the Clone Factory to see the entry fields that prepare it to be cloned to `S11ROGLD`.

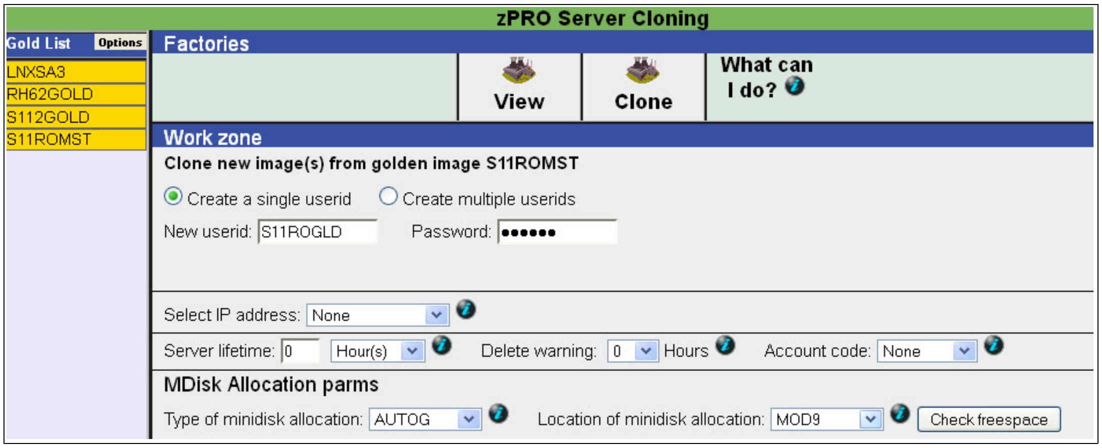


Figure 12-24 Cloning

5. A password for the new machine must be entered with minidisk allocation parms. No IP address parameters must be entered because the clone uses the same IP address as the gold image. If they are not both up at the same time, this is not a concern. The cloning operation allocates new minidisks from the Dirmaint group called `MOD9`. Click **Start Cloning** on the right. This creates the directory entry that is based on the old directory entry. The minidisks are created based on any read/write minidisks in the gold image and are copied from the gold image to the clone.

After the cloning operation is complete, `S11ROGLD` must be modified to replace the root (200) minidisk with a LINK to `S11ROMST` 200. When it is initially loaded, it uses a read-only version of the root file system.

The network device on the clone looks similar to the following example that shows that `eth1` is on our private 10 subnet:

```
s11rogld:/etc # ifconfig
eth1      Link encap:Ethernet  HWaddr 02:00:00:00:00:E5
          inet addr:10.1.1.101  Bcast:10.1.1.255  Mask:255.255.255.0
          inet6 addr: fe80::ff:fe00:e5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1492  Metric:1
          RX packets:263 errors:0 dropped:0 overruns:0 frame:0
          TX packets:234 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:21240 (20.7 Kb)  TX bytes:43457 (42.4 Kb)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:332 (332.0 b)  TX bytes:332 (332.0 b)
```

```
s11rogld:/etc # route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.1.1.100      0.0.0.0         UG    0      0        0 eth1
10.1.1.0        0.0.0.0         255.255.255.0   U     0      0        0 eth1
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
```

The TCP/IP stack on z/VM must also be configured to access the new subnet so that ZVPS can monitor each of the new Linux machines. The following example shows the modified GATEWAY section:

```
GATEWAY
; Network       Subnet          First           Link             MTU
; Address       Mask            Hop             Name             Size
; ------------- --------------- --------------- ---------------- -----
10.1.1.0/24                     9.12.7.5        OSA3020          1500
DEFAULTNET                      9.12.4.1        OSA3020          1500
; (End GATEWAY Static Routing information)
```

12.4 Shared Binary Oracle implementation: Oracle 11g

The original Linux image that was cloned contains a link to the LNXSA3 101 minidisk. This minidisk contains the Oracle 11gR2 installation files that can be used to trigger an installation of Oracle as new machines are created or at some point after the cloning is completed.

The reason for this is to significantly reduce the amount of required disk space because each virtual machine that is to install Oracle does not need a unique copy of the installation files.

Before the template machine is created, a mount point can be made to mount the new files, as shown in the following example:

```
# mkdir /opt/orainst
```

The following example shows the list of available disk devices:

```
s11rogld:/etc # lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   7042MB    1802880
0.0.0101   active      dasdb     94:4    ECKD  4096   21128MB   5409000
0.0.0103   active      dasdc     94:8    ECKD  4096   35MB      9000
0.0.0300   active      dasdd     94:12   FBA   512    256MB     524288
0.0.0301   active      dasde     94:16   FBA   512    512MB     1048576
0.0.0104   active      dasdf     94:20   ECKD  4096   35MB      9000
```

The following command makes the 101 minidisk, /dev/dasdb1, available for use:

```
# mount /dev/dasdb1 /opt/orainst
```

Oracle can then be installed.

12.5 Cloning 100 Oracle Servers for Development: Oracle 11g

zPro excels at many functions, including providing a graphical administration interface for z/VM. The ability to quickly create one or many virtual machines is another. Given a golden image, or template, zPro can copy that image as many times as required. The scenario in this section is of a development effort in which virtual machines are needed for large-scale testing of Oracle Database functions.

For more information about the installation of Oracle, see Appendix B, "Installing Oracle and creating a database 11.2.0.3 on Red Hat Enterprise Linux 6" on page 335. This section describes the cloning process that was used.

The read-only root function of SUSE Linux Enterprise Server 11 SP2 saves a significant amount of disk space. We allocate only two small minidisks for each clone that is created.

In the last section, a machine called `S11ROGLD` was created, which is the golden image that is used. Complete the following steps in zPro to add this image to the gold list.

1. Point to Manage Users and click **Directory Maintenance**.
2. Find `S11ROGLD` in the list, then drag it to the Add Gold factory.
3. From Manage Users, click **Cloning**. The list on the left should look as it does in Figure 12-25.

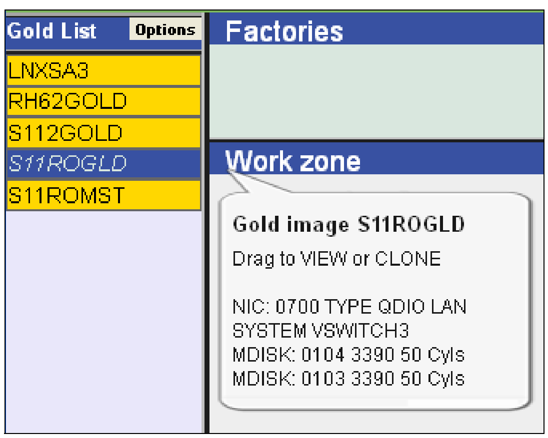


Figure 12-25 Add Gold

Moving the mouse over a gold image name displays the bubble that shows part of the virtual machine configuration. From this bubble, we can see that there are two 50-cylinder minidisks.

12.5.1 Cloning procedure

Complete the following steps to start cloning with zPro:

1. From Manage Users, click **Cloning**. Drag `S11ROGLD` to the Clone factory. The cloning form appears in the work area. The fields are completed as shown in Figure 12-26.

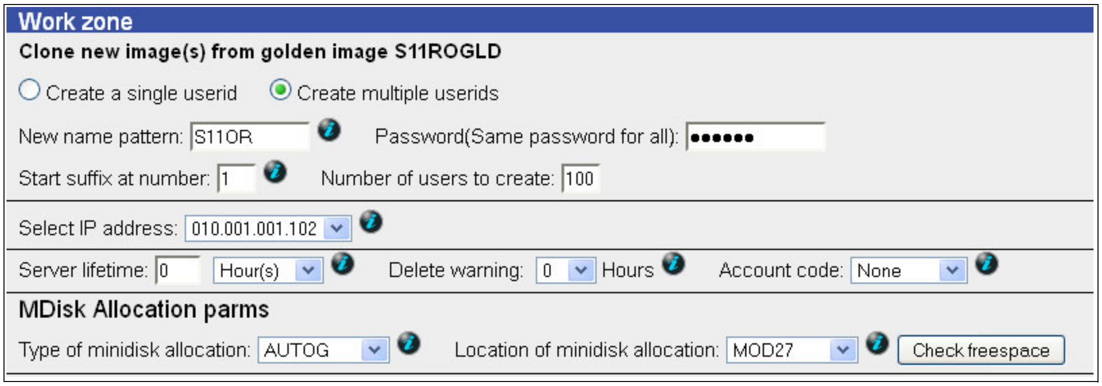


Figure 12-26 Cloning form

When you are cloning multiple virtual machines, you can enter the new name pattern in this form. It is the common characters for each new virtual machine. The remaining part of the name is an increasing numeric value, up to the number of machines that are created. The name pattern must allow for all of the possible numbers to be generated. Because 100 machines are cloned, the pattern must be five or fewer characters. The pattern `S110R` was chosen in our example.

The next field is the CP (or ESM) password that is used for all of the new machines.

Next is the starting value that is to be generated for the first clone, then the number of clones to be created.

The next section is where the IP address is selected for the first clone. IP addresses are assigned automatically to subsequent clones from that point in the list.

The next section determines the lifetime of the cloned images. A value entered here does expire (and delete) the clones after the time frame that is indicated. An optional expiration email can be delivered, which indicates that the clones are about to expire. The server life time can be changed in the Manage Users menu section of the zPro Server Expirations window.

The minidisks for the clones were put in our `MOD27` group, which is made up of 3390 Model 27 devices. The last section of the cloning form shows the group that was selected. To see how much disk space is available in that group, click **Check Freespace**. The results are shown in Figure 12-27 on page 304. It shows that there is plenty of disk space for 100 copies of 100 cylinders of minidisk

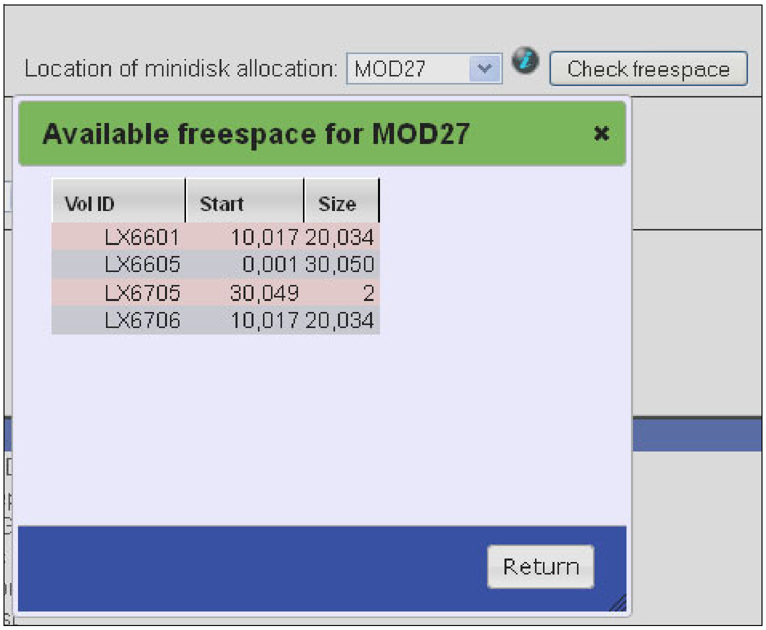


Figure 12-27 Check freespace

2. When you are satisfied that the values are entered correctly, click **Start Cloning** on the right side of the form. As shown in Figure 12-28, the action log (which appears at the bottom of the window) shows the current step of the cloning process for each new virtual machine.

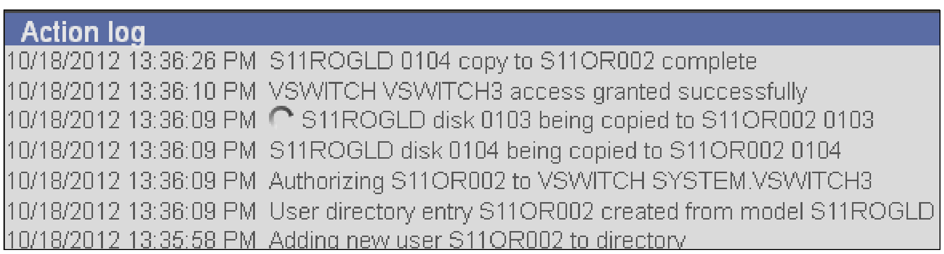


Figure 12-28 Start cloning action log

A window opens that shows the clone that is in the process of being created, as shown in Figure 12-29.

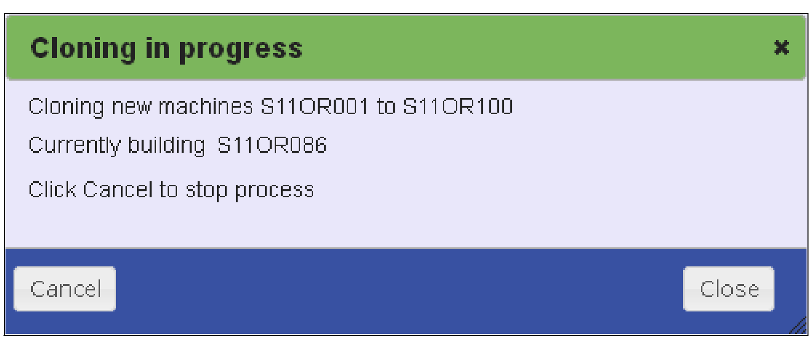


Figure 12-29 Cloning in progress

After all of the clones are created, the user status display shows them all as Inactive, as shown in Figure 12-30.

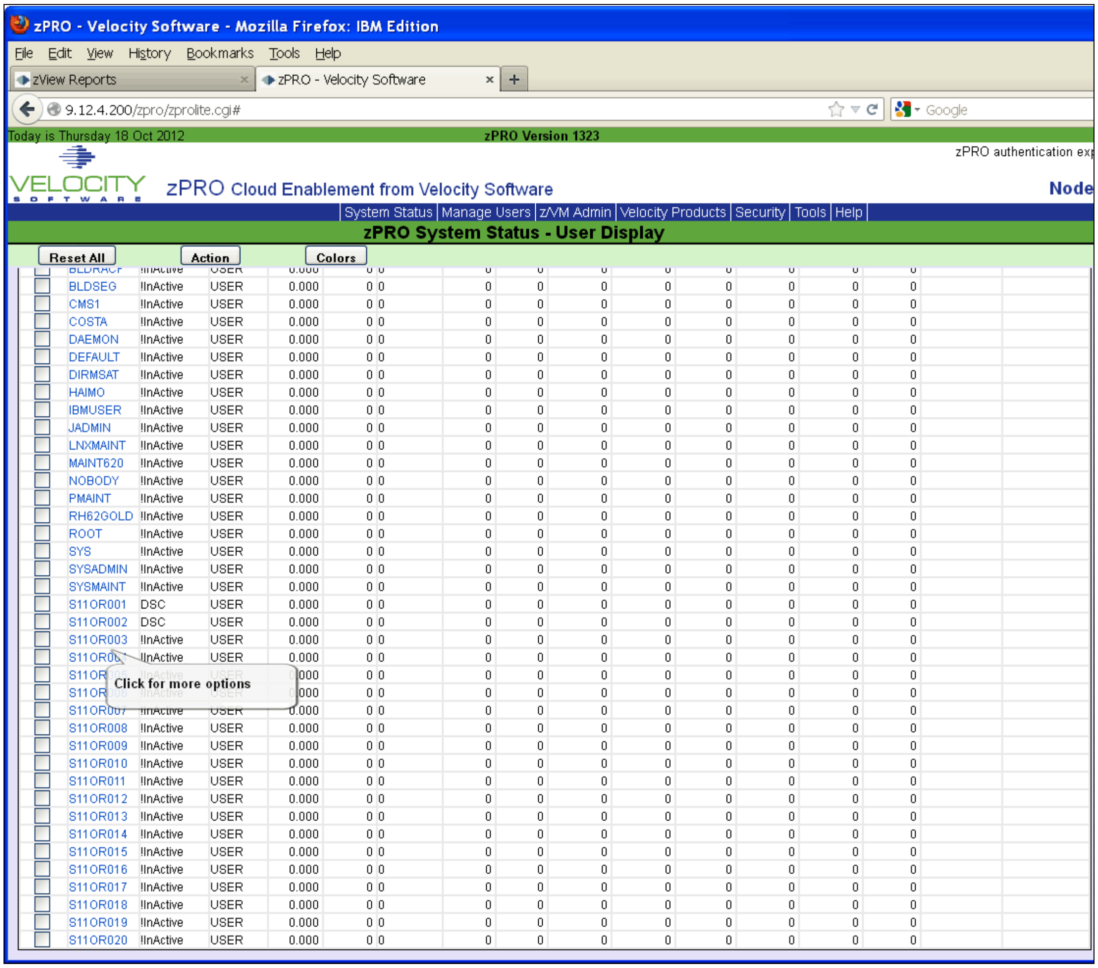


Figure 12-30 Inactive servers

3. Click a virtual machine name to open the window that is shown in Figure 12-31.

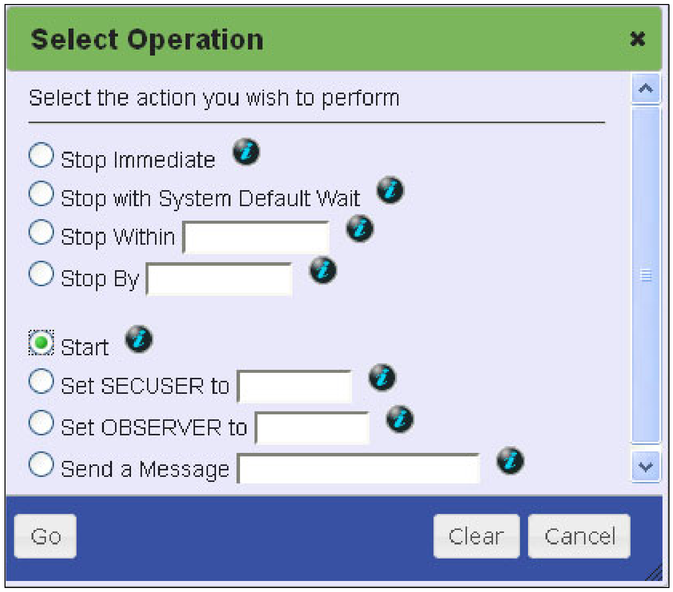


Figure 12-31 Select Operation window

4. Select the **Start** option, then click **Go** to activate (`XAUTOLOG`) the virtual machine. The Alert window opens, as shown in Figure 12-32, in which the start is confirmed.

Figure 12-32 Activate/XAUTOLOG Server

5. To activate many virtual machines, select the check boxes to the left of the machine names that must be activated, then click **Action** at the top of the list to open the same window. Click **Start** → **Go**.

As shown in Figure 12-33, many of the virtual machines are activated. The column to the right of the machine name shows `DSC`, which indicates that it is running and is disconnected.

☐	SFSZVPS	DSC	USER	0.001	64	100R	331	330
☐	S11OR001	DSC	USER	0.013	1024	50R	18,004	17,997
☐	S11OR002	DSC	USER	0.013	1024	50R	20,134	20,128
☐	S11OR003	DSC	USER	0.015	1024	50R	21,308	21,302
☐	S11OR004	DSC	USER	0.013	1024	50R	16,913	16,896
☐	S11OR005	DSC	USER	0.013	1024	50R	22,716	22,709
☐	S11OR006	DSC	USER	0.014	1024	50R	17,489	17,483
☐	S11OR007	DSC	USER	0.014	1024	50R	15,785	15,778
☐	S11OR008	DSC	USER	0.013	1024	50R	16,591	16,584
☐	S11OR009	DSC	USER	0.013	1024	50R	17,053	17,036
☐	S11OR010	DSC	USER	0.013	1024	50R	15,344	15,337
☐	S11OR011	DSC	USER	0.013	1024	50R	20,608	20,601
☐	S11OR012	DSC	USER	0.014	1024	50R	14,798	14,792
☐	S11OR013	DSC	USER	0.014	1024	50R	20,739	20,733
☐	S11OR014	DSC	USER	0.015	1024	50R	20,266	20,259
☐	S11OR015	DSC	USER	0.013	1024	50R	19,450	19,444
☐	S11OR016	DSC	USER	0.014	1024	50R	19,687	19,681
☐	S11OR017	DSC	USER	0.014	1024	50R	21,651	21,644
☐	S11OR018	DSC	USER	0.014	1024	50R	18,962	18,945
☐	S11OR019	DSC	USER	0.004	1024	50R	14,749	14,743
☐	S11OR020	DSC	USER	0.004	1024	50R	14,991	14,984
☐	S11OR021	DSC	USER	0.014	1024	50R	20,905	20,899
☐	S11OR022	DSC	USER	0.014	1024	50R	20,512	20,506
☐	S11OR023	DSC	USER	0.014	1024	50R	17,062	17,045
☐	S11OR024	DSC	USER	0.014	1024	50R	21,545	21,539
☐	S11OR025	DSC	USER	0.017	1024	50R	21,113	21,106
☐	S11OR026	DSC	USER	0.014	1024	50R	17,046	17,039
☐	S11OR027	DSC	USER	0.014	1024	50R	17,366	17,360
☐	S11OR028	DSC	USER	0.013	1024	50R	15,510	15,503
☐	S11OR029	DSC	USER	0.015	1024	50R	16,433	16,416
☐	S11OR030	DSC	USER	0.017	1024	50R	21,164	21,157
☐	S11OR031	DSC	USER	0.014	1024	50R	20,743	20,736
☐	S11OR032	DSC	USER	0.015	1024	50R	20,815	20,809
☐	S11OR033	DSC	USER	0.014	1024	50R	20,478	20,471
☐	S11OR034	DSC	USER	0.014	1024	50R	16,410	16,393
☐	S11OR035	DSC	USER	0.018	1024	50R	21,983	21,977
☐	S11OR036	DSC	USER	0.013	1024	50R	20,810	20,803

Figure 12-33 Activated servers

12.6 References

The following documents and books were used as reference material in writing this chapter:

- *z/VM V6R2 Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6190-02
- *z/VM V6R2 Systems Management Application Programming*, SC24-6234-03
- *z/VM and Linux on IBM System z: The Cloud Computing Cookbook for z/VM 6.2 RHEL 6.2 and SLES11 SP2*
- *SUSE Linux Enterprise Server 11 SP2 Release Notes*, which is available at this website:

  https://www.suse.com/releasenotes/x86_64/SUSE-SLES/11-SP2/
- *Sharing and maintaining Linux under z/VM*, REDP-4322

Part 4

Appendixes

We included in this publication the following appendices to provide information that might be useful when you set up your Oracle database on Linux on System z:

- Appendix A, “Setting up Red Hat Enterprise Linux 6.3 for Oracle” on page 311 for an Oracle Database. Oracle products require that more rpms are installed.
- Appendix B, “Installing Oracle and creating a database 11.2.0.3 on Red Hat Enterprise Linux 6” on page 335. The following Redpapers™ are available about how to install Oracle Real Application Clusters on Linux on System z:
  - http://www.redbooks.ibm.com/abstracts/redp4788.html?Open
  - http://www.redbooks.ibm.com/abstracts/redp9131.html?Open
- Appendix C, “Working effectively with Oracle support” on page 349. There is a dedicated team working on this platform.
- Appendix D, “Additional material” on page 357.

A

Setting up Red Hat Enterprise Linux 6.3 for Oracle

Attention: Linux commands that are running as root are prefixed with `#` in this appendix.

This appendix describes the steps to install a Red Hat Enterprise Linux 6.3 (RHEL 6.3) guest to support an 11gR2 Database and other Oracle products. We assume that you are performing a new installation, not updating a Red Hat Enterprise Linux 5 and an existing Oracle database because a new installation is recommended.

Note: Oracle certified Oracle Database 11.2.0.3 on Red Hat Enterprise Linux 6.2 and above in 1Q 2013. This chapter is based on installing RHEL 6.3.

Oracle 11gR2(11.2.0.3) is certified on RHEL 6.2 and above. For RHEL 6.3, errata 1156 should be installed. For more information, see this website:

http://rhn.redhat.com/errata/RHSA-2012-1156.html

The following methods are available to use to install Red Hat Enterprise Linux 6:

- A GUI installation of RHEL 6.3 Base code and then `oravalidate` rpm is run, which brings in all the missing rpms that are required to install Oracle Database.
- An installation with a kickstart file that runs a silent installation of RHEL 6.

This appendix included the following topics:

- Introduction
- Step 1: Starting the Red Hat bootstrap loader
- Step 2: Installing Red Hat Enterprise Linux
- Step 3: Running oravalidate rpm to import all other rpms
- Installing and setting up vncserver
- Step 4: Customizing the Linux setup for Oracle
- Summary

A.1 Introduction

Before you begin this process, ensure that z/VM guest Directory entries were prepared and the user can log in to z/VM and use CMS. The Red Hat installation process includes the following two major steps, and there is a third step to add the rpms that is needed for Oracle:

1. Starting the Red Hat bootstrap loader
2. Installing Red Hat Enterprise Linux
3. Running `oravalidate` rpm to bring in all the other rpms

These steps are described next.

A.2 Step 1: Starting the Red Hat bootstrap loader

This process starts the Red Hat bootstrap loader system and includes the following tasks:

- Defining the Linux guest
- Defining the PARM and CONF files
- Defining the EXEC file
- Run the EXEC file:
  - Punching and loading the Red Hat reader images
  - Connecting to the installation images (this installation uses NFS)
  - Making the Virtual Network Computing (VNC) connection to perform the next Red Hat Enterprise Linux 6.3 stage of the installation

These steps are described next.

A.2.1 Defining the Linux guest

The installation requires that the guest has a defined network interface. The network interface is defined by using the **`DEFINE NIC`** and **`COUPLE`** statements, as shown in Example A-1.

Example A-1 Dirmaint entry for Linux guest

```
USER PAZXXT10 IBMPASS 4G 4G BEG 64
   ACCOUNT C0000620 LINUX
   CPU 00 BASE
   CPU 01
   IPL CMS PARM AUTOCR
   IUCV *IDENT GATEANY GATEWAY REVOKE
   IUCV ALLOW
   OPTION TODENABLE LNKNOPAS
   POSIXINFO UID 996
   CONSOLE 001F 3215 A
   SPOOL 000C 2540 READER *
   SPOOL 000D 2540 PUNCH A
   SPOOL 000E 1403
   /*Setup network devices*/
   'CP DEFINE NIC 420 QDIO'
   'CP COUPLE 480 SYSTEM' vsw

*
```

```
   LINK MAINT 019E 019E RR
   LINK MAINT 019D 019D RR
   LINK MAINT 0190 0190 RR
   LINK VMLINUX 0104 0191 RR
   MDISK 0200 3390 1 10016 LXD20D MR ROOT
*  swap disks
   MDISK 0207 FB-512 V-DISK 4194296 MR
*  local disks
```

A.2.2 Defining the PARM and CONF files

The installation needs kernel parameters and network definitions to complete successfully. The following methods are available:

- Defining the parameters dynamically as you perform the installation, as described in A.4, “Step 3: Running oravalidate rpm to import all other rpms” on page 326.
- Create a PARM file that containing kernel parameters and a CONF file, which contains network and disk parameters. Though not required, this is recommended. For more information, see Example A-2 and Example A-3.

Example A-2 Generic PARM file

```
/*Sample RHEL 6.3 PARM file*/
ROOT=/DEV/RAM0 RO IP=OFF RAMDISK_SIZE=40000 RO IP=OFF CMSDASD=191
CMSDASD=191 CMSCONFFILE=RHU3.CONF
```

Example A-3 RHU3 CONF file

```
/*Sample RHEL63 CONF file:*/
DASD=200,207
HOSTNAME=PAZXXT10.US.ORACLE.COM
NETTYPE=qeth
IPADDR=130.35.52.18
SUBCHANNELS=0.0.0480,0.0.0481,0.0.0482
NETMASK=255.255.252.0
SEARCHDNS=US.ORACLE.COM
GATEWAY=130.35.52.1
DNS=130.35.249.41
MTU=8192
PORTNAME=UNASSIGNED
LAYER2=0
```

Some values should be changed to work in your environment; however, the overall format of the CONF file should not change. The `SUBCHANNELS` parameter defines the subchannel addresses for the NIC. `LAYER2=0` is used because the VSWITCH is operating in Layer 3 (IP) mode. If the VSWITCH is operating in Layer 2 (ETH) mode, you should set `LAYER2=1` and `VSWITCH=1`. If you are unsure, you should check with the network administrator. For more information about this parameter, see this website:

http://kbase.redhat.com/faq/FAQ_69_12554.shtm

A.2.3 Defining the EXEC File

In this section, we assume that the Red Hat Enterprise Linux 6.3 installation tree is available via FTP. From z/VM, log in as the user and transfer the kernel and initial RAMdisk image (`initrd`) that is necessary to begin the installation. Be sure to set the logical record length to 80 before transferring the kernel and `initrd` (**`QUOTE LOCSITE FIX 80`** if transferring via FTP from z/VM, or **`SITE FIX 80`** if transferring via FTP to z/VM).

Next, create the EXEC that is shown in Example A-4, then run it to begin the installation.

Example A-4 Example Punch file: Redhat exec

```
/*EXEC TO PUNCH RHEL6.3 TO THE RDR  */
'CL RDR'
'PURGE RDR ALL'
'SPOOL PUNCH * RDR'
'PUNCH KERNEL IMG A (NOH'
'PUNCH GENERIC PRM A (NOH'
'PUNCH INITRD IMG A (NOH'
'CH RDR ALL KEEP NOHOLD'
'I 00C'
```

The commands associated punch (loads into the reader) the necessary images in the correct order and prepares them to be loaded. The last command initially loads the reader, which loads the files that were punched.

When the initial program load (IPL) is run, the reader is loaded (as shown in Example A-5) and the Linux guest is ready to load.

Example A-5 RDRLIST

```
PAZXXT11 RDRLIST S0 V 164 Trunc+164 Size=3 Line=1 Col=1 Alt=0
Cmd Filename Filetype Class User at   Node   Hold Records Date  Time
     KERNEL  IMG      PUN A PAXZXXT11 HQCMS2 NONE 113488 11/09 13:02.28
     GENERIC PRM      PUN A PAXZXXT11 HQCMS2 NONE      2 11/09 13:02.33
     INITRD  IMG      PUN A PAXZXXT11 HQCMS2 NONE 223040 11/09 13:02.33
```

Because the CONF file contains the networking and DASD information, the installation proceeds in silent mode, which brings up Figure A-6 on page 317.

A.2.4 Completing the first step of the installation process

You are now logged on to the CMS guest to run the `redhat exec` file. It punches the files to the reader and loads the Linux installation. You see many messages as it checks and configures the disks. At the end of the process, you are instructed to log on by using root ID. We found it best to use an SSH client like PuTTY to connect to the Linux and use the installation ID.

The 191 disk should have the five files that are shown in Example A-6.

Example A-6 Output of 191 disk

```
GENERIC  PRM      A1          V         53         2          1 11/30/12  8:19
INITRD   IMG      A1          F         80    226772       4430 11/12/12  9:45
KERNEL   IMG      A1          F         80    113591       1723 11/12/12  9:44
REDHAT   EXEC     A1          V         38         9          1 11/26/12 17:26
RHU3     CONF     A1          F         80        12          1 11/26/12 17:26
```

The installation process takes you through several windows, as shown in Figure A-1, Figure A-2 on page 315, Figure A-3 on page 316, Figure A-4 on page 316, and Figure A-5 on page 317.

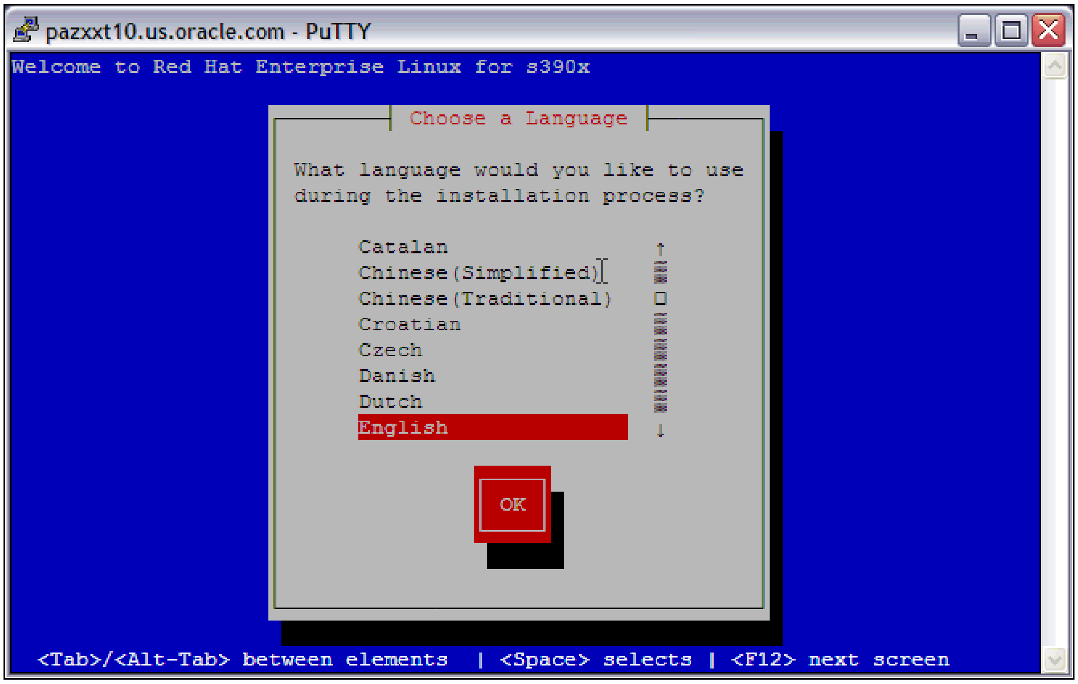


Figure A-1 Choose the language to use during Red Hat Installation

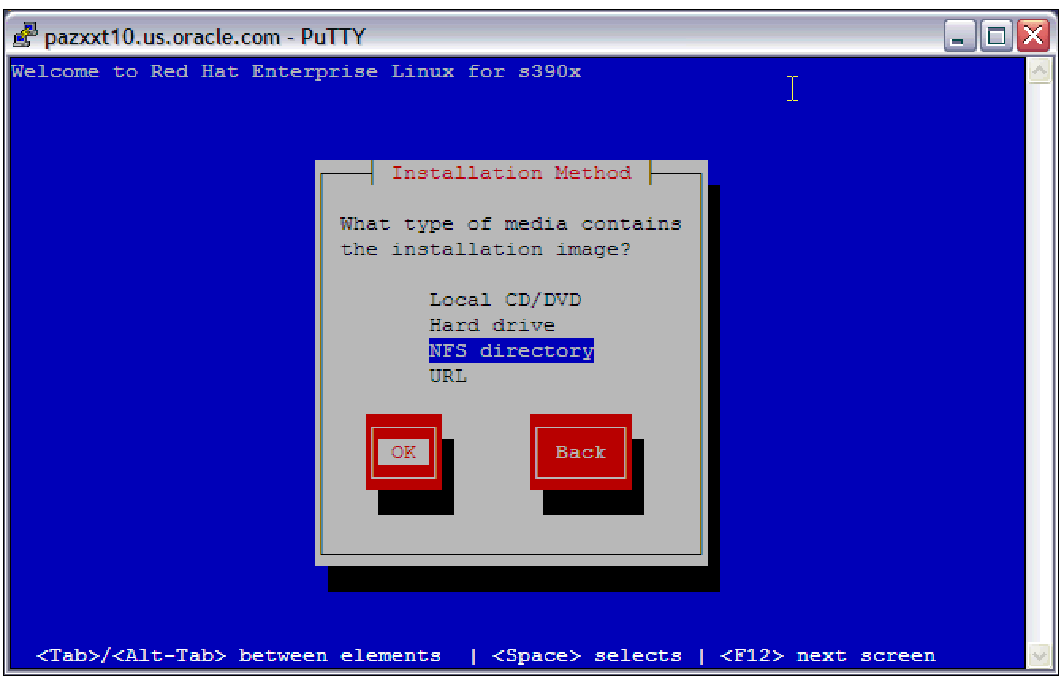


Figure A-2 Identify the media that contains Red Hat 6.3 code

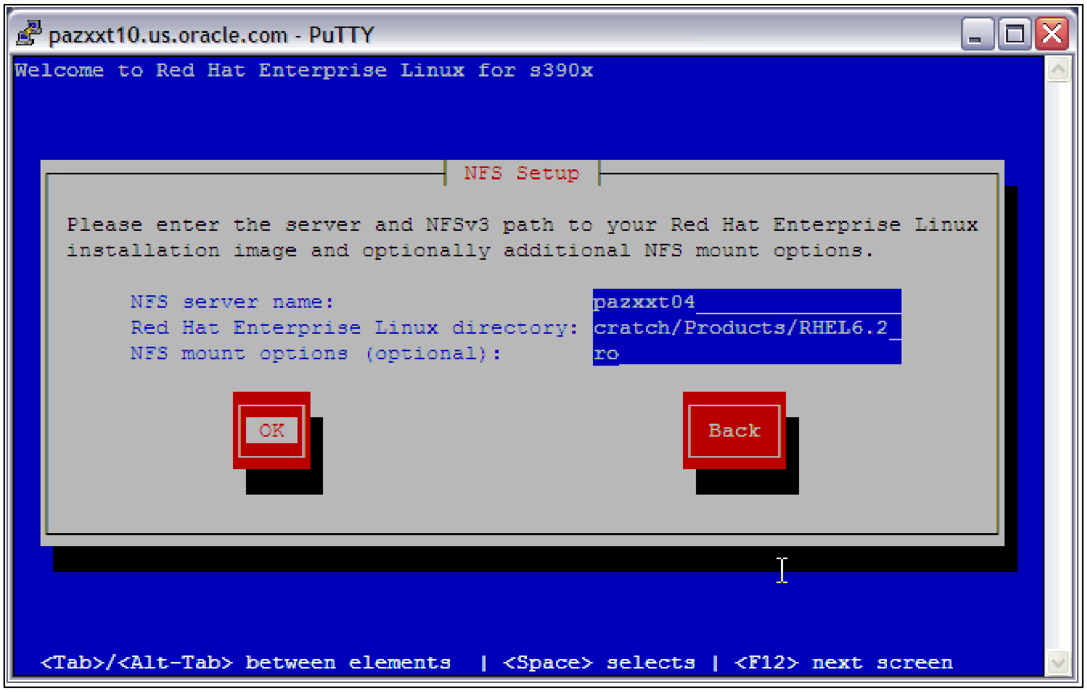


Figure A-3 NFS setup location

Figure A-4 Request to start VNC for a GUI interface

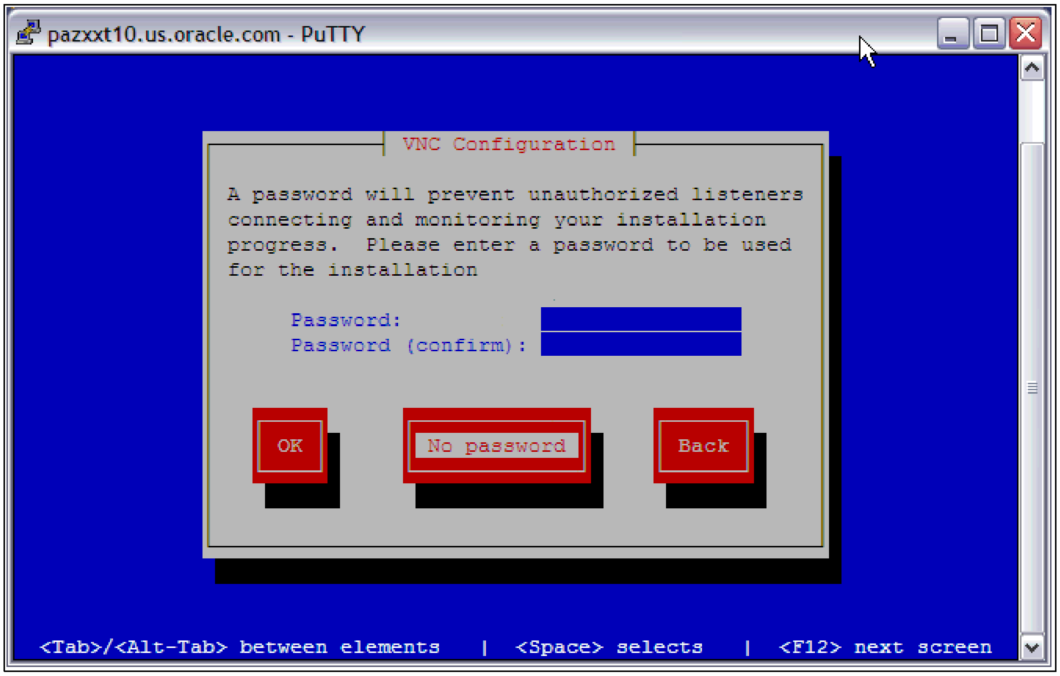


Figure A-5 Set the password for VNC

The GUI installation process is now started.

A.3 Step 2: Installing Red Hat Enterprise Linux

The Red Hat Enterprise Linux 6.3 installation system is started from the bootstrap process and displays the image that is shown in Figure A-6.

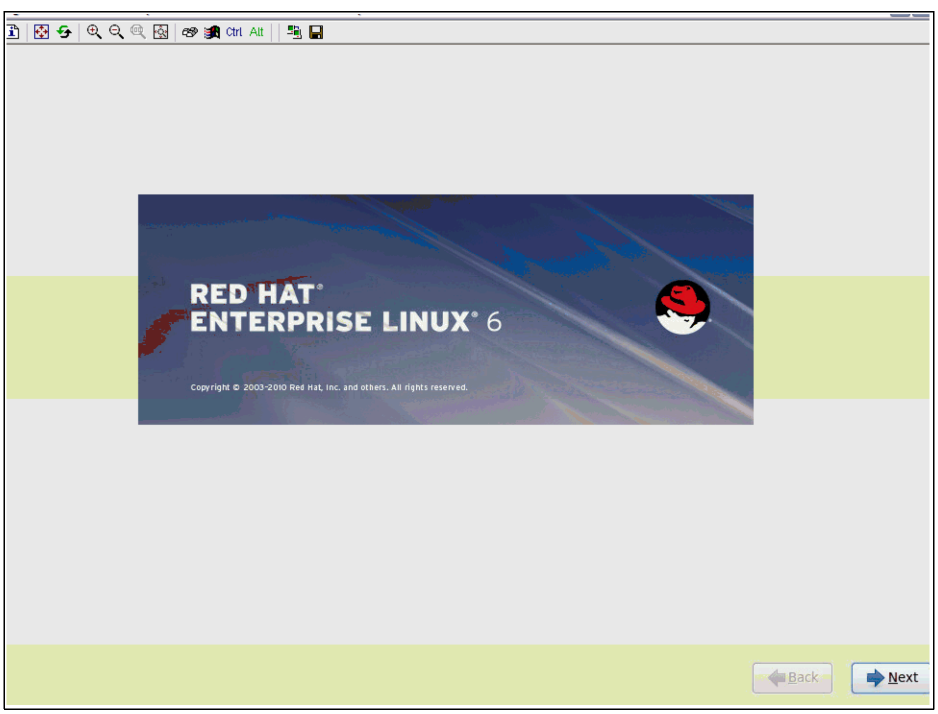


Figure A-6 Initial Red Hat window

We selected the Specialized Storage Devices option as shown in Figure A-7.

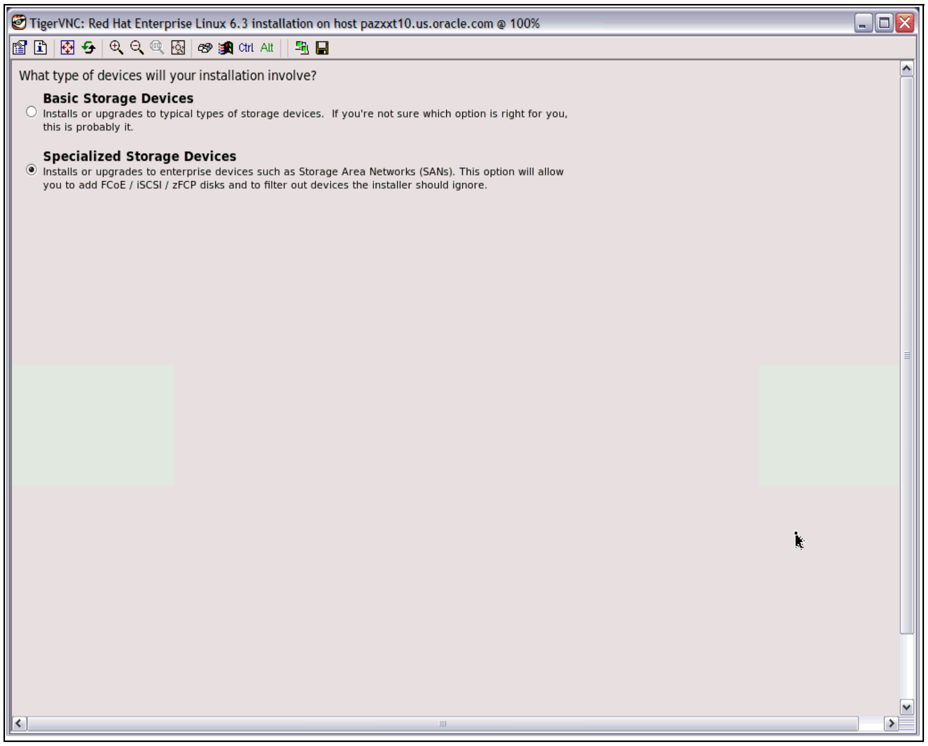


Figure A-7 Storage

We selected the two available DASD disks to use for storage, as shown in Figure A-8.

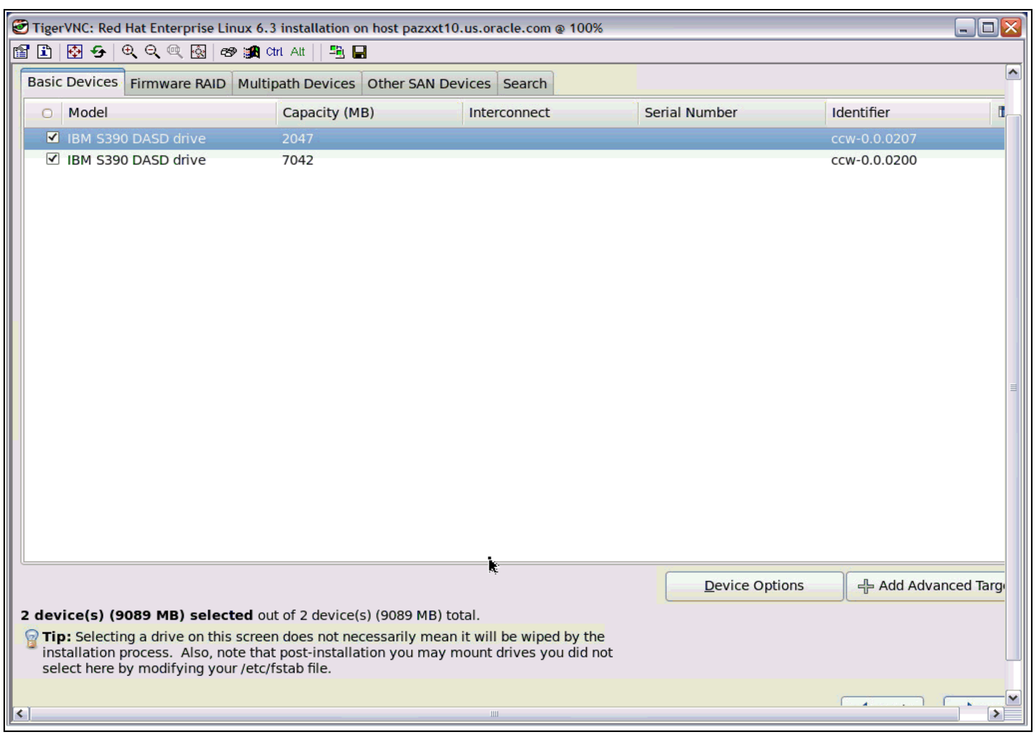


Figure A-8 Storage for RHEL

These disks were used before, so we confirmed that we wanted to write over the existing data, as shown in Figure A-9 on page 319.

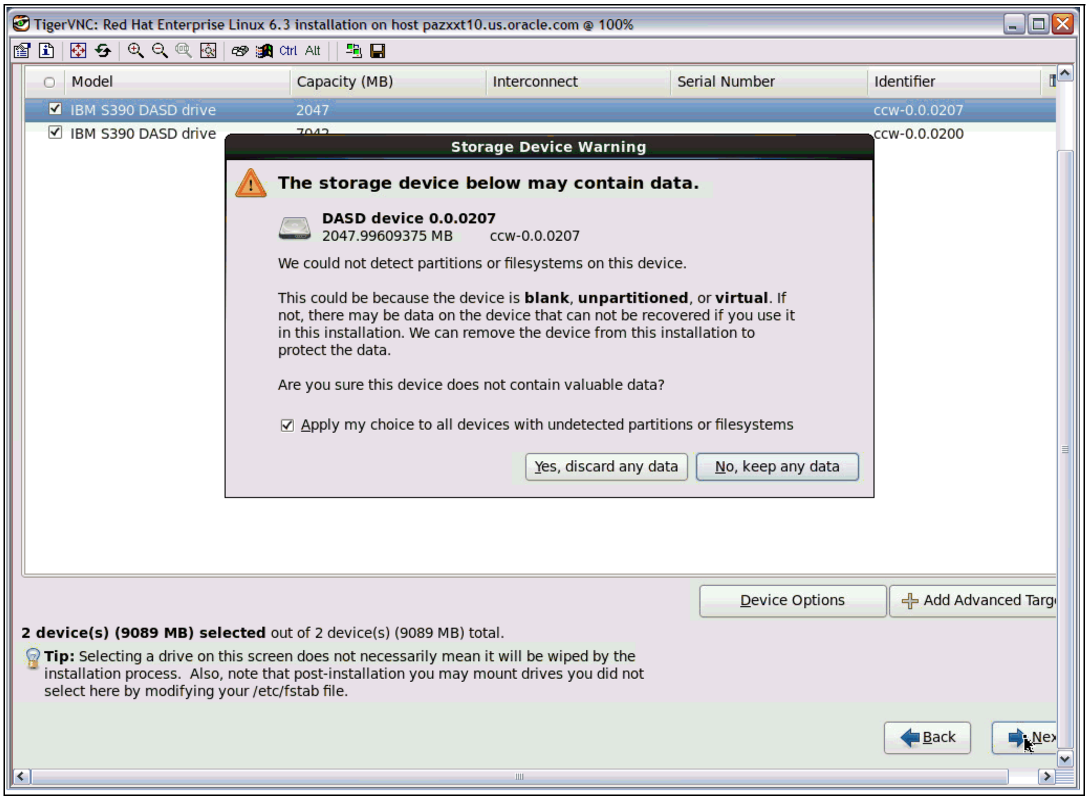


Figure A-9 Storage message

Enter the host name to be used by the Linux guest, as shown in Figure A-10 on page 320.

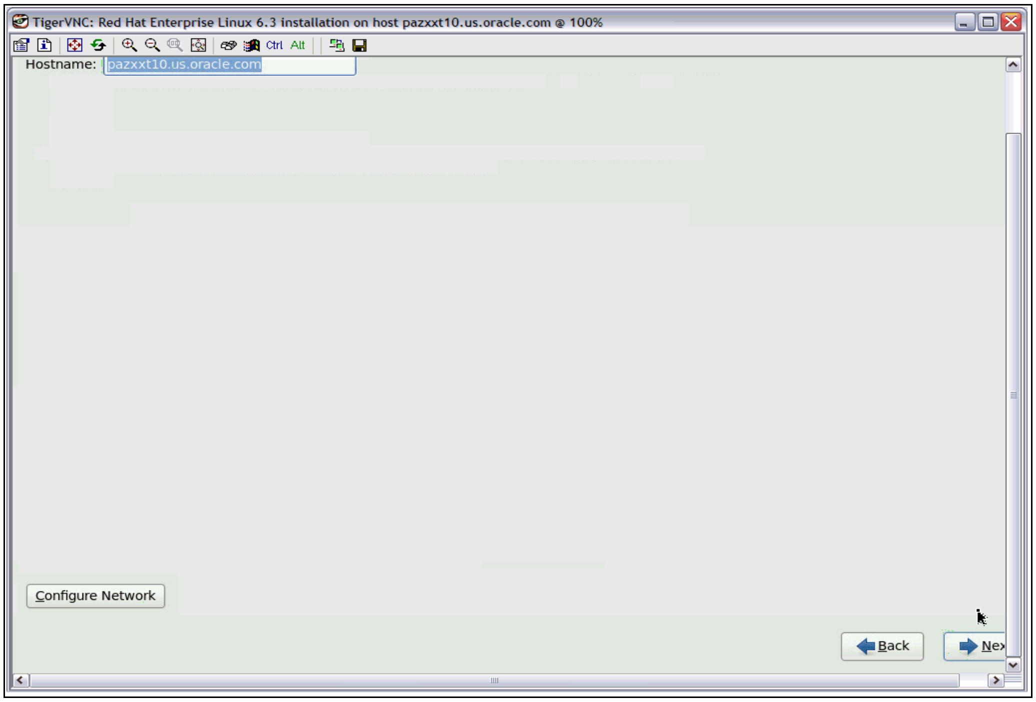


Figure A-10 Host name

In our case, it detected a previous installation. We chose **Fresh Installation**, as shown in Figure A-11.

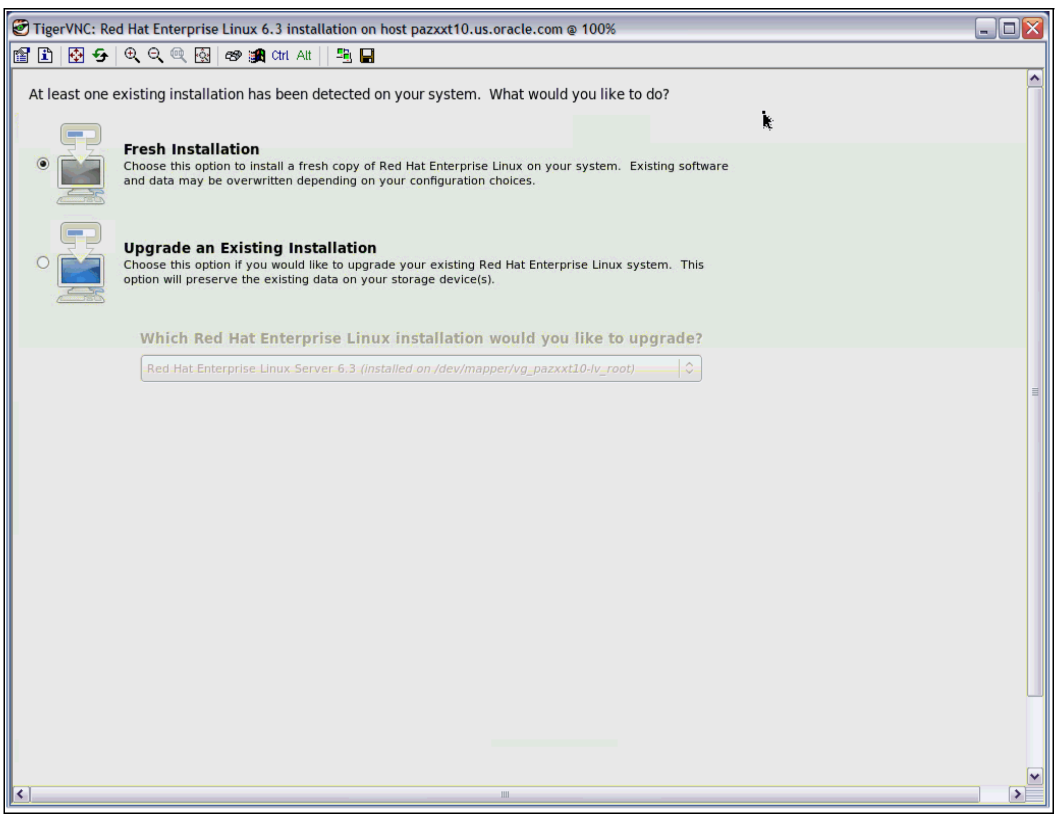


Figure A-11 Fresh installation or upgrade

We then chose **Pacific Time Zone**, as shown in Figure A-12.

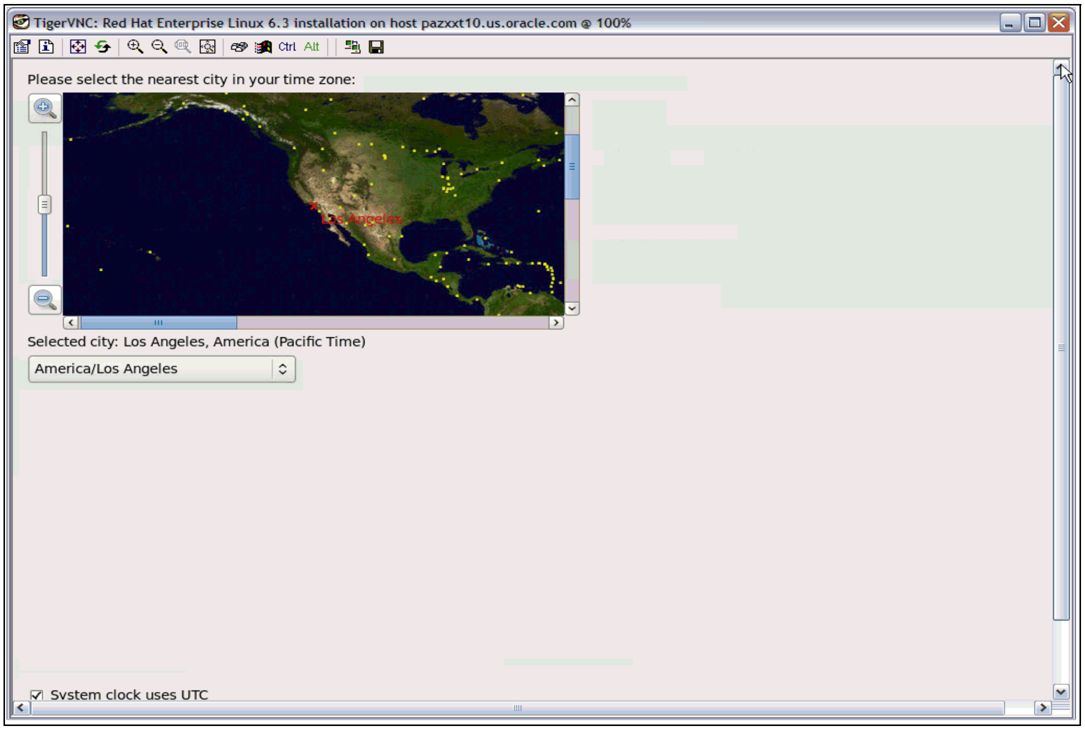


Figure A-12 Time zone choice

We choose the Fresh Installation, as shown in Figure A-13.

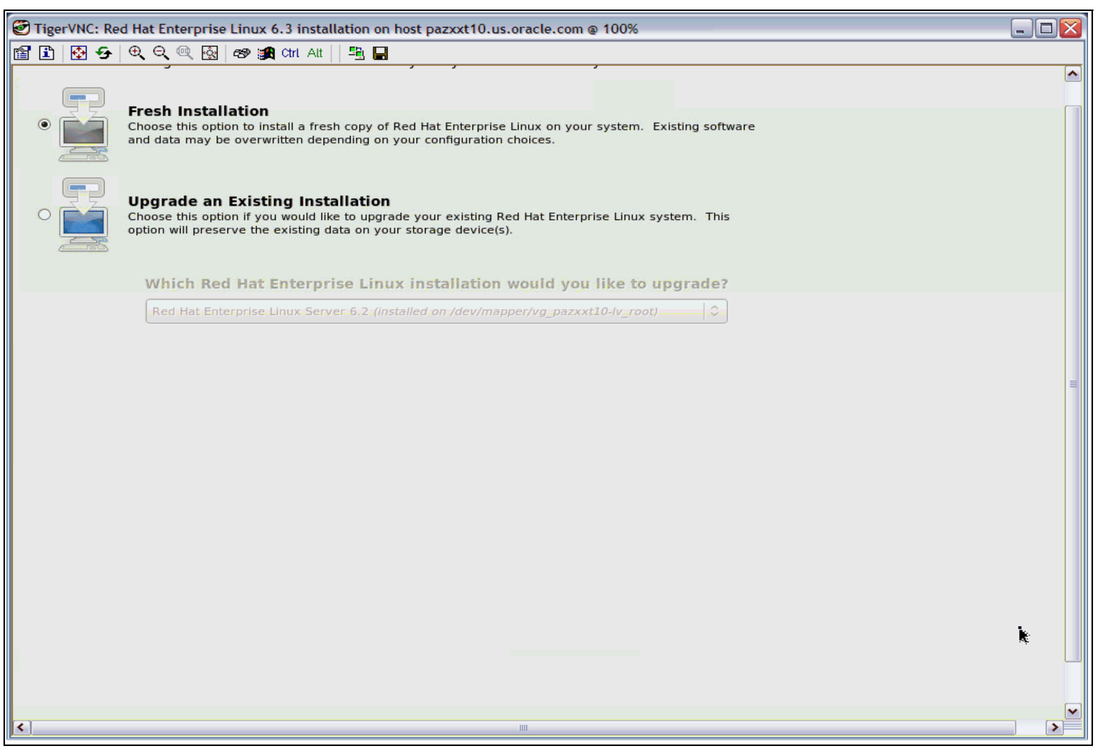


Figure A-13 Fresh Installation of RHEL

Choose your password and make sure that it is noted, as shown in Figure A-14.

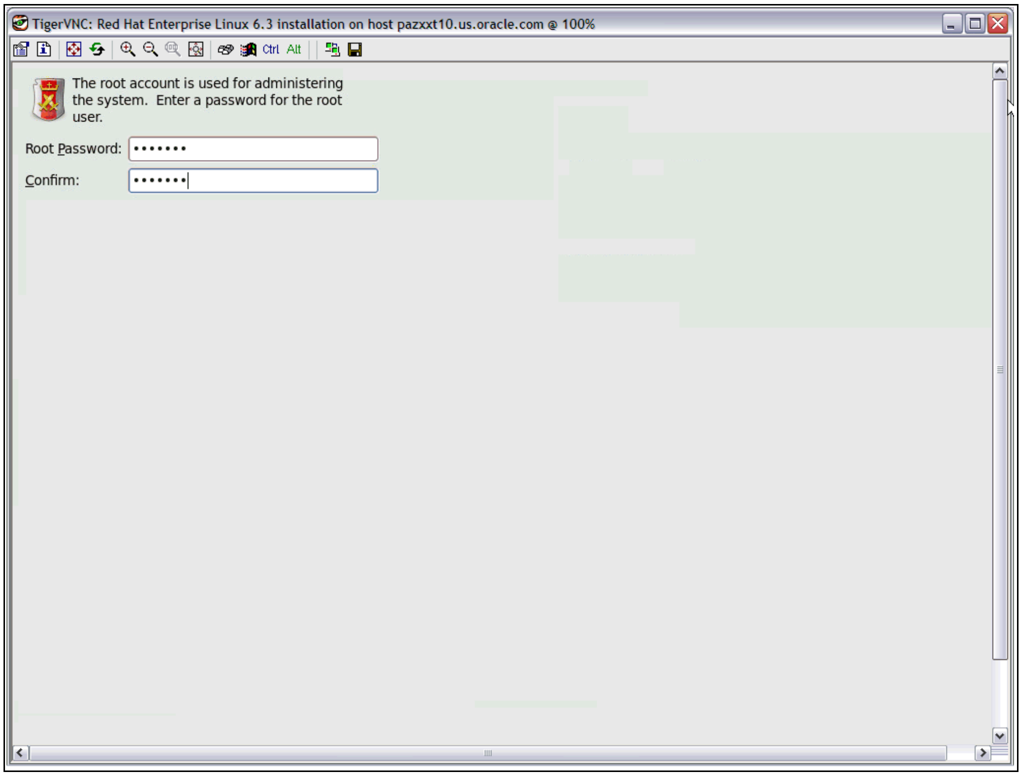


Figure A-14 Root account password

Figure A-15 shows the chosen type of installation.

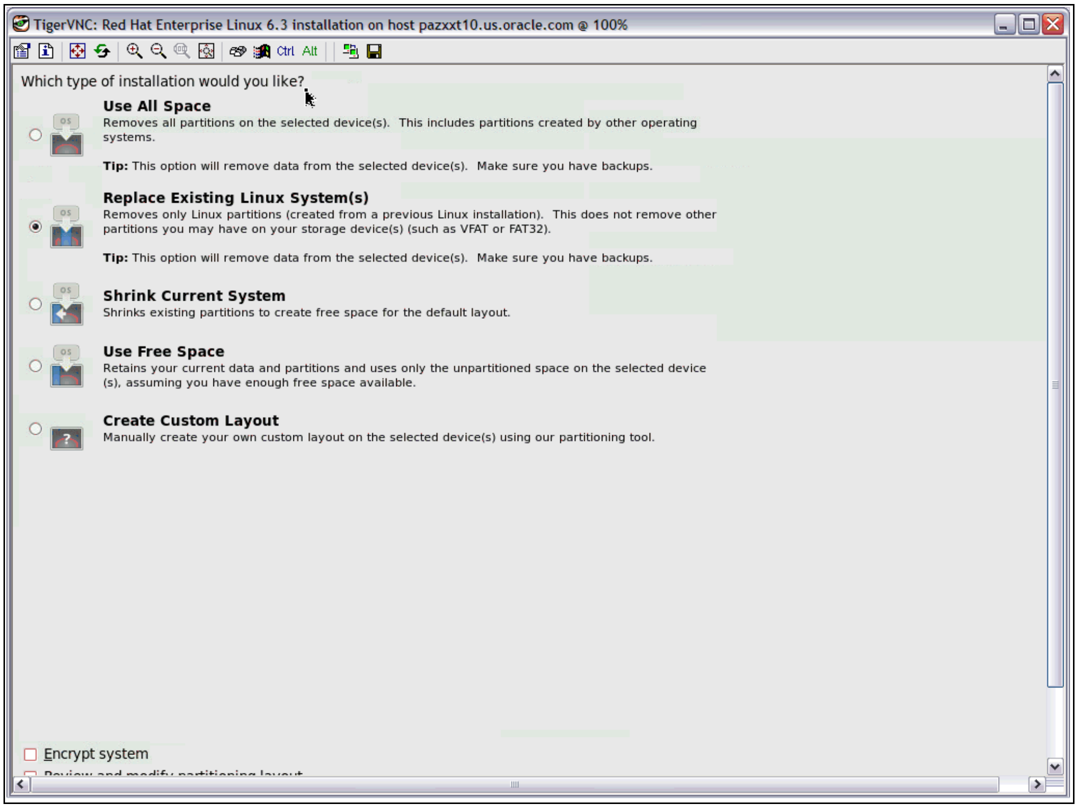


Figure A-15 Type of Installation

We choose the target DASDs for the installation, as shown in Figure A-16.

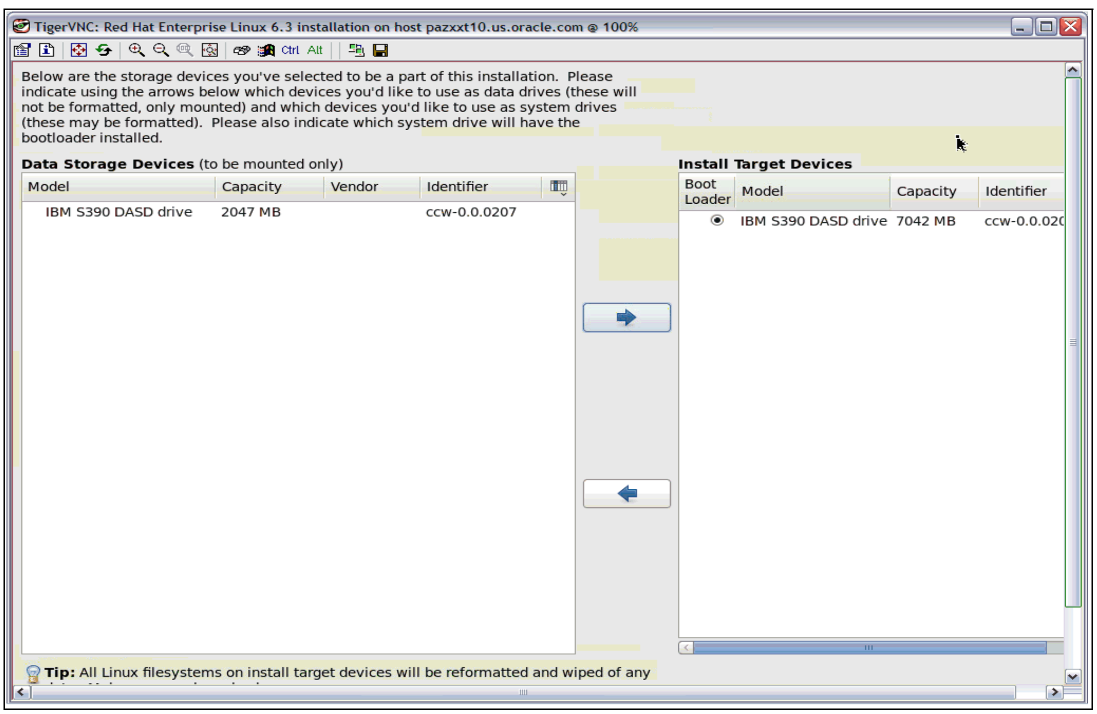


Figure A-16 Data Storage

The disks are now formatted and an LVM is created, as shown in Figure A-17.

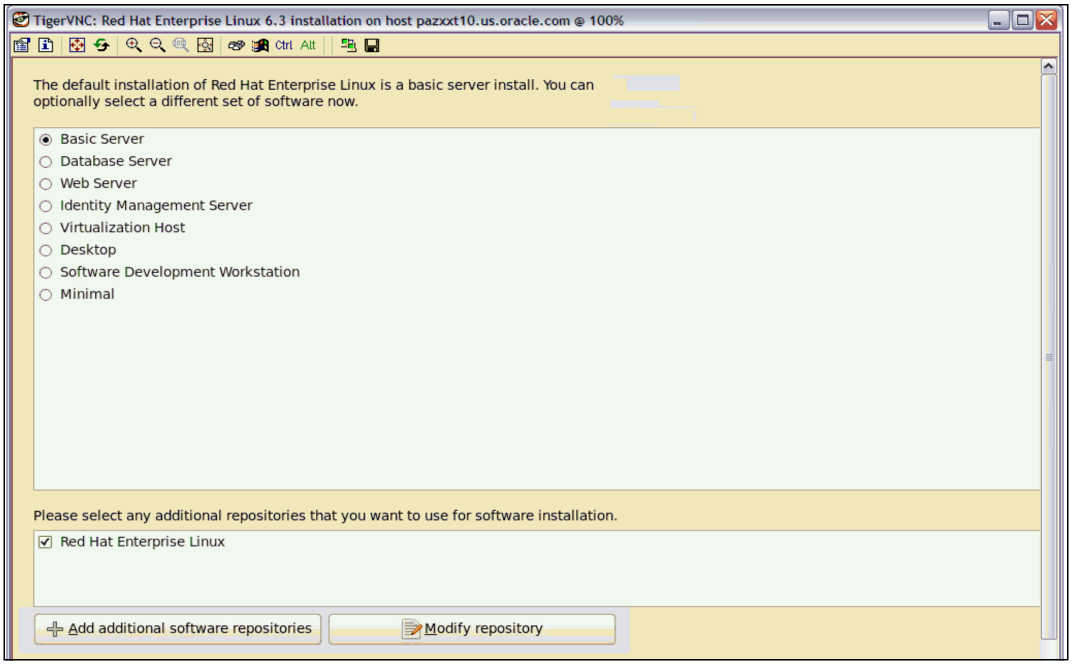


Figure A-17 Choosing Base Install Only

We choose **Basic Server** installation and added the other rpms for Oracle later by using the `oravalidate` process, as shown in Figure A-18.

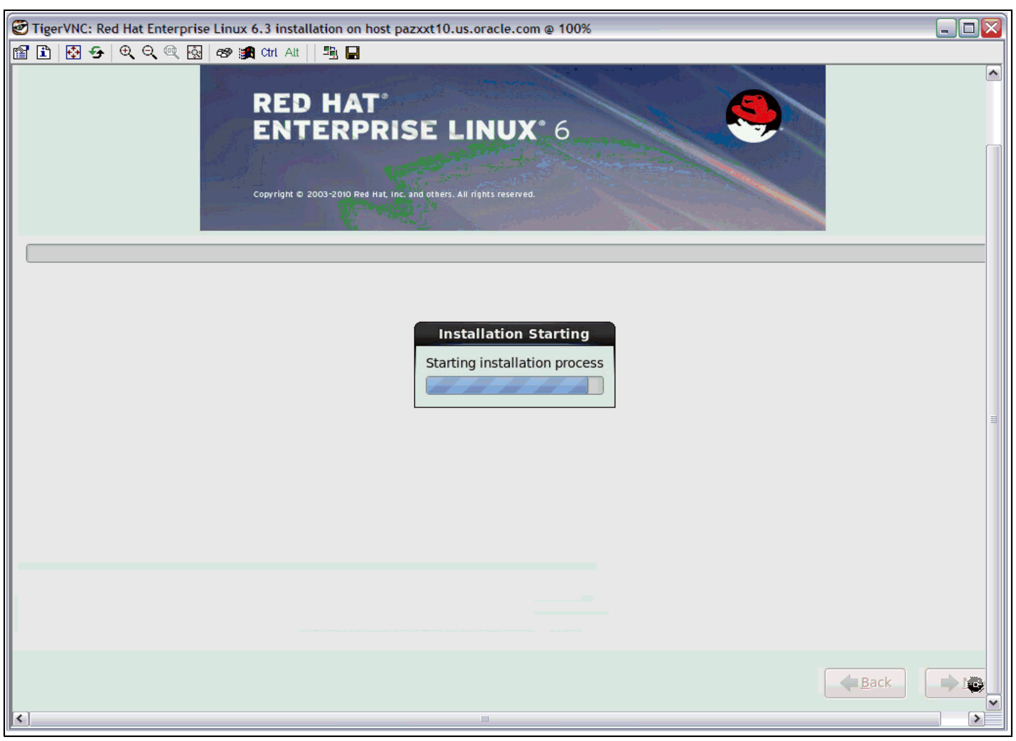


Figure A-18 Installation in Process

This stage takes 10 - 20 minutes to complete. You see the packages as they are installed, as shown in Figure A-19.

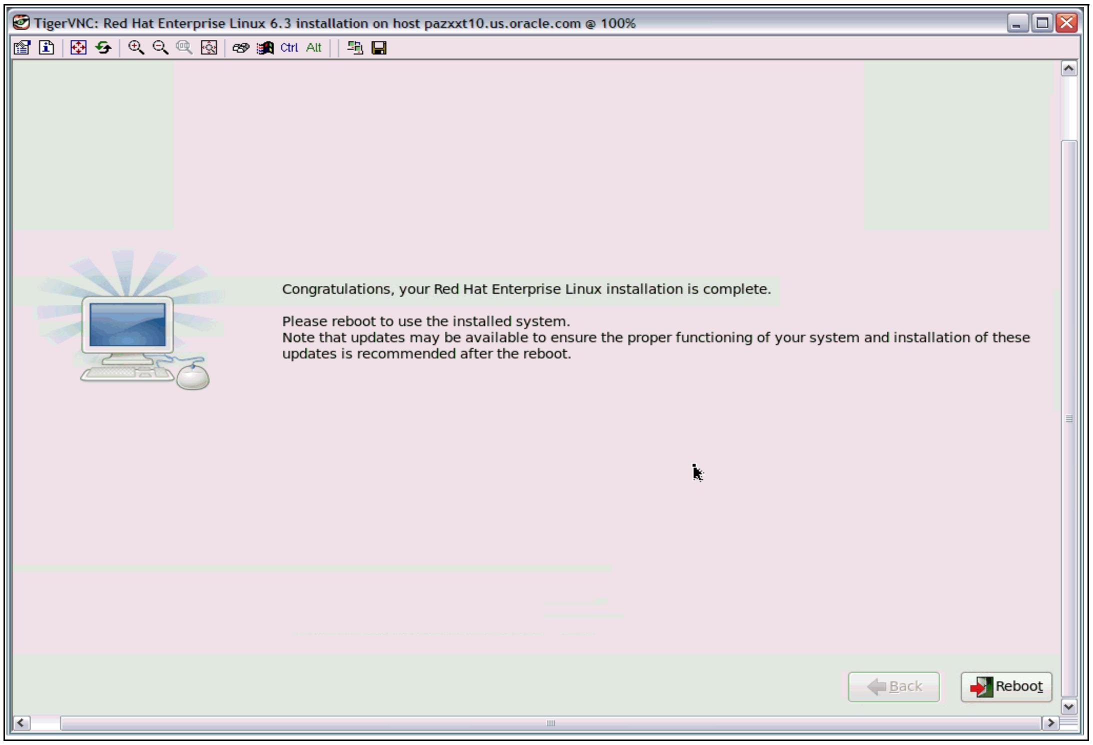


Figure A-19 Base installation is complete

After you rebooted Linux guest, you should install errata that are recommended for Red Hat 6.3 on Linux on System z, which is available at this website:

http://rhn.redhat.com/errata/RHSA-2012-1156.html

An example of the errata is shown in Example A-7.

Example A-7 Example of the installation of the errata

```
Packages:
kernel-2.6.32-279.9.1.el6.s390x.rpm
kernel-firmware-2.6.32-279.9.1.el6.noarch.rpm
kernel-devel-2.6.32-279.9.1.el6.s390x.rpm
kernel-headers-2.6.32-279.9.1.el6.s390x.rpm

Install:

[root@pazxxt10 RHEL6.3]# rpm -ivh  kernel-devel-2.6.32-279.9.1.el6.s390x.rpm
warning: kernel-devel-2.6.32-279.9.1.el6.s390x.rpm: Header V3 RSA/SHA256
Signature, key ID fd431d51: NOKEY
Preparing...                ########################################### [100%]
   1:kernel-devel           ########################################### [100%]
[root@pazxxt10 RHEL6.3]# rpm -ivh  kernel-headers-2.6.32-279.9.1.el6.s390x.rpm
warning: kernel-headers-2.6.32-279.9.1.el6.s390x.rpm: Header V3 RSA/SHA256
Signature, key ID fd431d51: NOKEY
Preparing...                ########################################### [100%]
   1:kernel-headers         ########################################### [100%]
[root@pazxxt10 RHEL6.3]# rpm -ivh  kernel-firmware-2.6.32-279.9.1.el6.noarch.rpm
```

```
warning: kernel-firmware-2.6.32-279.9.1.el6.noarch.rpm: Header V3 RSA/SHA256
Signature, key ID fd431d51: NOKEY
Preparing...                ########################################### [100%]
   1:kernel-firmware        ########################################### [100%]
[root@pazxxt10 RHEL6.3]# rpm -ivh  kernel-2.6.32-279.9.1.el6.s390x.rpm
warning: kernel-2.6.32-279.9.1.el6.s390x.rpm: Header V3 RSA/SHA256 Signature, key
ID fd431d51: NOKEY
Preparing...                ########################################### [100%]
   1:kernel                 ########################################### [100%]
[root@pazxxt10 RHEL6.3]#
```

A.3.1 Next step

The base installation is now complete, but there are other rpms that are needed to prepare this system for the Oracle Database installation.

The next section describes how to use the `ora-validate` rpm to do this. After the reboot, you log on as the `root` user.

A.4 Step 3: Running oravalidate rpm to import all other rpms

In this step, you prepare the Linux guest with all the RHEL 6.3 rpms that are needed by Oracle. For information about how to use and obtain the latest copy of `ora-validate`, see the My Oracle Support note.

The following commands are used:

- Create a directory:

```
# mkdir /redhat
# chmod 755 /redhat
```

- Make the following changes to point to the Redhat source directory:

```
# vi /etc/yum.repos.d/rhel-source.repo
[root@pazxxt10 yum.repos.d]# vi rhel-source.repo
[rhel-source]
name=Red Hat Enterprise Linux $releasever - $basearch - Source
baseurl=file:///redhat/zlinux/RHEL6/U2/Server/
enabled=1
gpgcheck=0
gpgkey=file:///redhat/zlinux/RHEL6/U2/Server/RPM-GPG-KEY-redhat-release
```

- Mount RHEL 6.3 media:

```
# mount.nfs lhotse:/software/redhat /redhat
```

- Update the repository file:

```
# rpm --import RPM-GPG-KEY-redhat-release
```

A.4.1 Command to run ora-validate and the output

You must download the `oravalidate` rpm from My Oracle Support. After you extract the file, run it by using the **`yum install`** command, as shown in Example A-8.

Lines of output are produced as the other needed rpms are installed.

Example A-8 Running oravalidate

```
# yum install /root/ora-val-rpm-EL6-DB-11.2.0.3-1.s390x.rpm
   ====================================================
   Loaded plugins: product-id, security, subscription-manager
   Updating certificate-based repositories.
   rhel-source                                                  | 4.0 kB
   00:00 ...
   rhel-source/primary_db                                       | 2.5 MB
   00:01 ...
   Setting up Install Process
   Examining /root/ora-val-rpm-EL6-DB-11.2.0.3-1.s390x.rpm:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   Marking /root/ora-val-rpm-EL6-DB-11.2.0.3-1.s390x.rpm to be installed
   Resolving Dependencies
   --> Running transaction check
   ---> Package ora-val-rpm-EL6-DB.s390x 0:11.2.0.3-1 will be installed
   --> Processing Dependency: compat-libcap1 >= 1.10-1 for package:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: compat-libstdc++-33 >= 3.2.3-69 for package:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: compat-libstdc++-33(s390-64) >= 3.2.3-69 for
   package: ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: compat-libstdc++-33(s390-32) >= 3.2.3-69 for
   package: ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: elfutils-libelf-devel >= 0.152-1 for package:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: gcc >= 4.4.5-6 for package:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: gcc-c++ >= 4.4.5-6 for package:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: glibc(s390-32) >= 2.12-1.25 for package:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: ksh >= 20100621-6 for package:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: libaio(s390-32) >= 0.3.107-10 for package:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: libgcc(s390-32) >= 4.4.5-6 for package:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: libstdc++(s390-32) >= 4.4.5-6 for package:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: glibc-headers >= 2.12-1.25 for package:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x

--> Processing Dependency: glibc-devel(s390-64) >= 2.12-1.25 for package: ora-

   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: glibc-devel(s390-32) >= 2.12-1.25 for package:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: libstdc++-devel >= 4.4.5-6 for package:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: libstdc++-devel(s390-64) >= 4.4.5-6 for package:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
   --> Processing Dependency: libstdc++-devel(s390-32) >= 4.4.5-6 for package:
   ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
```

```
--> Processing Dependency: libaio-devel(s390-64) >= 0.3.107-10 for package:
ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
--> Processing Dependency: libaio-devel(s390-32) >= 0.3.107-10 for package:
ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
--> Processing Dependency: compat-libstdc++-33(s390-64) for package:
ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
--> Processing Dependency: glibc-devel(s390-64) for package:
ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
--> Processing Dependency: libstdc++-devel(s390-64) for package:
ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
--> Processing Dependency: libaio-devel(s390-64) for package:
ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
--> Running transaction check
---> Package compat-libcap1.s390x 0:1.10-1 will be installed
---> Package compat-libstdc++-33.s390 0:3.2.3-69.el6 will be installed
---> Package compat-libstdc++-33.s390x 0:3.2.3-69.el6 will be installed
---> Package elfutils-libelf-devel.s390x 0:0.152-1.el6 will be installed
---> Package gcc.s390x 0:4.4.6-3.el6 will be installed
--> Processing Dependency: cpp = 4.4.6-3.el6 for package: gcc-4.4.6-3.el6.s390x
--> Processing Dependency: cloog-ppl >= 0.15 for package: gcc-4.4.6-3.el6.s390x
---> Package gcc-c++.s390x 0:4.4.6-3.el6 will be installed
--> Processing Dependency: libmpfr.so.1()(64bit) for package:
gcc-c++-4.4.6-3.el6.s390x
---> Package glibc.s390 0:2.12-1.47.el6 will be installed
--> Processing Dependency: libfreebl3.so(NSSRAWHASH_3.12.3) for package:
glibc-2.12-1.47.el6.s390
--> Processing Dependency: libfreebl3.so for package: glibc-2.12-1.47.el6.s390
---> Package glibc-devel.s390 0:2.12-1.47.el6 will be installed
---> Package glibc-devel.s390x 0:2.12-1.47.el6 will be installed
---> Package glibc-headers.s390x 0:2.12-1.47.el6 will be installed
--> Processing Dependency: kernel-headers >= 2.2.1 for package:
glibc-headers-2.12-1.47.el6.s390x
--> Processing Dependency: kernel-headers for package:
glibc-headers-2.12-1.47.el6.s390x
---> Package ksh.s390x 0:20100621-12.el6 will be installed
---> Package libaio.s390 0:0.3.107-10.el6 will be installed
---> Package libaio-devel.s390 0:0.3.107-10.el6 will be installed
---> Package libaio-devel.s390x 0:0.3.107-10.el6 will be installed
---> Package libgcc.s390 0:4.4.6-3.el6 will be installed
---> Package libstdc++.s390 0:4.4.6-3.el6 will be installed
---> Package libstdc++-devel.s390 0:4.4.6-3.el6 will be installed
---> Package libstdc++-devel.s390x 0:4.4.6-3.el6 will be installed
--> Running transaction check
---> Package cloog-ppl.s390x 0:0.15.7-1.2.el6 will be installed
--> Processing Dependency: libppl.so.7()(64bit) for package:
cloog-ppl-0.15.7-1.2.el6.s390x
--> Processing Dependency: libppl_c.so.2()(64bit) for package:
cloog-ppl-0.15.7-1.2.el6.s390x
---> Package cpp.s390x 0:4.4.6-3.el6 will be installed
---> Package kernel-headers.s390x 0:2.6.32-220.el6 will be installed
---> Package mpfr.s390x 0:2.4.1-6.el6 will be installed
---> Package nss-softokn-freebl.s390 0:3.12.9-11.el6 will be installed
--> Running transaction check
---> Package ppl.s390x 0:0.10.2-11.el6 will be installed
--> Finished Dependency Resolution
```

```
Dependencies Resolved
================================================================================
Package             Arch  Version       Repository Size
================================================================================
Installing:
 ora-val-rpm-EL6-DB  s390x 11.2.0.3-1     /ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
0.0
Installing for dependencies:
 cloog-ppl           s390x 0.15.7-1.2.el6 rhel-source 89 k
 compat-libcap1      s390x 1.10-1         rhel-source 18 k
 compat-libstdc++-33 s390  3.2.3-69.el6   rhel-source 182 k
 compat-libstdc++-33 s390x 3.2.3-69.el6   rhel-source 186 k
 cpp                 s390x 4.4.6-3.el6    rhel-source 3.2 M
 elfutils-libelf-devel s390x 0.152-1.el6    rhel-source 31 k
 gcc                 s390x 4.4.6-3.el6    rhel-source 6.5 M
 gcc-c++             s390x 4.4.6-3.el6    rhel-source 4.2 M
 glibc               s390  2.12-1.47.el6  rhel-source 3.8 M
 glibc-devel         s390  2.12-1.47.el6  rhel-source 973 k
 glibc-devel         s390x 2.12-1.47.el6  rhel-source 974 k
 glibc-headers       s390x 2.12-1.47.el6  rhel-source 593 k
 kernel-headers      s390x 2.6.32-220.el6 rhel-source 1.6 M
 ksh                 s390x 20100621-12.el6 rhel-source 709 k
 libaio              s390  0.3.107-10.el6 rhel-source 21 k
 libaio-devel        s390  0.3.107-10.el6 rhel-source 13 k
 libaio-devel        s390x 0.3.107-10.el6 rhel-source 13 k
 libgcc              s390  4.4.6-3.el6    rhel-source 85 k
 libstdc++           s390  4.4.6-3.el6    rhel-source 306 k
 libstdc++-devel     s390  4.4.6-3.el6    rhel-source 1.6 M
 libstdc++-devel     s390x 4.4.6-3.el6    rhel-source 1.6 M
 mpfr                s390x 2.4.1-6.el6    rhel-source 162 k
 nss-softokn-freebl  s390  3.12.9-11.el6  rhel-source 156 k
 ppl                 s390x 0.10.2-11.el6  rhel-source 1.2 M


Transaction Summary
================================================================================
Install      25 Package(s)

Total download size: 28 M
Installed size: 79 M
Is this ok [y/N]:
Downloading Packages:
--------------------------------------------------------------------------------
Total                                                  27 MB/s |  28 MB
00:01
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : libstdc++-devel-4.4.6-3.el6.s390x 1/25
  Installing : elfutils-libelf-devel-0.152-1.el6.s390x
2/25
  Installing : kernel-headers-2.6.32-220.el6.s390x
3/25
```

```
  Installing : libgcc-4.4.6-3.el6.s390
4/25
  Installing : nss-softokn-freebl-3.12.9-11.el6.s390
5/25
  Installing : glibc-2.12-1.47.el6.s390
6/25
  Installing : glibc-headers-2.12-1.47.el6.s390x
7/25
  Installing : glibc-devel-2.12-1.47.el6.s390
8/25
  Installing : mpfr-2.4.1-6.el6.s390x
9/25
  Installing : libaio-0.3.107-10.el6.s390
10/25
  Installing : libstdc++-4.4.6-3.el6.s390
11/25
  Installing : libstdc++-devel-4.4.6-3.el6.s390
12/25
  Installing : libaio-devel-0.3.107-10.el6.s390
13/25
  Installing : libaio-devel-0.3.107-10.el6.s390x
14/25
  Installing : cpp-4.4.6-3.el6.s390x
15/25
  Installing : glibc-devel-2.12-1.47.el6.s390x
16/25
  Installing : compat-libstdc++-33-3.2.3-69.el6.s390x
17/25
  Installing : compat-libcap1-1.10-1.s390x
18/25
  Installing : ksh-20100621-12.el6.s390x
19/25
  Installing : ppl-0.10.2-11.el6.s390x
20/25
  Installing : cloog-ppl-0.15.7-1.2.el6.s390x
21/25
  Installing : gcc-4.4.6-3.el6.s390x
22/25
  Installing : gcc-c++-4.4.6-3.el6.s390x
23/25
  Installing : compat-libstdc++-33-3.2.3-69.el6.s390
24/25
  Installing : ora-val-rpm-EL6-DB-11.2.0.3-1.s390x
25/25
***************************************************************************
*     Validation complete - please install any missing rpms               *
*   The following output should display both (s390) - 31-bit and          *
*       (s390x) 64-bit rpms - Please provide the output to Oracle         *
*            Support If you are still encountering problems.              *
***************************************************************************
Found          glibc-dev (s390)
Found          glibc-dev (s390x)
Found          libaio (s390)
Found          libaio (s390x)
Found          libaio-devel (s390)
```

```
   Found          libaio-devel (s390x)
   Found          compat-libstdc++-33 (s390)
   Found          compat-libstdc++-33 (s390x)
   Found          glibc (s390)
   Found          glibc (s390x)
   Found          libgcc (s390)
   Found          libgcc (s390x)
   Found          libstdc++ (s390)
   Found          libstdc++ (s390x)
   Found          libstdc++-devel (s390)
   Found          libstdc++-devel (s390x)
   Found          libaio-dev (s390)
   Found          libaio-dev (s390x)
rhel-source/productid                                        | 1.7 kB     00:00
...
Installed products updated.
Installed:
  ora-val-rpm-EL6-DB.s390x 0:11.2.0.3-1
Dependency Installed:
    cloog-ppl.s390x 0:0.15.7-1.2.el6       compat-libcap1.s390x 0:1.10-1
     compat-libstdc++-33.s390 0:3.2.3-69.el6  compat-libstdc++-33.s390x
   0:3.2.3-69.el6
     cpp.s390x 0:4.4.6-3.el6                  elfutils-libelf-devel.s390x
   0:0.152-1.el6
    gcc.s390x 0:4.4.6-3.el6               gcc-c++.s390x 0:4.4.6-3.el6
    glibc.s390 0:2.12-1.47.el6            glibc-devel.s390 0:2.12-1.47.el6
    glibc-devel.s390x 0:2.12-1.47.el6     glibc-headers.s390x 0:2.12-1.47.el6
    kernel-headers.s390x 0:2.6.32-220.el6   ksh.s390x 0:20100621-12.el6
    libaio.s390 0:0.3.107-10.el6          libaio-devel.s390 0:0.3.107-10.el6
    libaio-devel.s390x 0:0.3.107-10.el6    libgcc.s390 0:4.4.6-3.el6
    libstdc++.s390 0:4.4.6-3.el6          libstdc++-devel.s390 0:4.4.6-3.el6
    libstdc++-devel.s390x 0:4.4.6-3.el6    mpfr.s390x 0:2.4.1-6.el6
    nss-softokn-freebl.s390 0:3.12.9-11.el6  ppl.s390x 0:0.10.2-11.el6


   Complete!
```

The Oracle required rpms are installed and you can now prepare for the Oracle installation.

A.5 Installing and setting up vncserver

You need vncserver installed so you have a GUI interface for the Oracle Universal Installer.

We installed tigervnc that is included with RHEL 6.3 by using the following command:

```
# yum -y install tigervnc-server openmotif xterm xsetroot xorg-x11-xauth
```

You then must update the vncserver system configuration file by running the following commands:

```
# cd /etc/sysconfig/
# vi vncservers
   # VNCSERVERS="1:root"
   # VNCSERVERARGS[2]="-geometry 800x600 -nolisten tcp -localhost"
```

You uncomment by removing the # sign and then adding other user IDs, as shown in the following example:

```
VNCSERVERS=”2:oracle”
```

Save the file.

If you enabled the firewall, then must update the IP tables by editing the iptables file, as shown in the following example:

```
# vi iptables
```

Add the following lines

```
A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 5801 -j ACCEPT
A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 5802 -j ACCEPT
```

Then, you must recycle iptables by running the following commands:

```
# service iptables stop
# service iptables start
# service vncserver start
```

You need a password to access vncserver from your desktop. If you are logged in as root, enter vncpasswd to create the password, as shown in the following example:

```
Command  vncpasswd
Password:
Verify:
xauth:  creating new authority file /root/.Xauthority

New 'pazxxt10.us.oracle.com:1 (root)' desktop is pazxxt10.us.oracle.com:1
Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/pazxxt10.us.oracle.com:1.log
```

A.6 Step 4: Customizing the Linux setup for Oracle

You should now make the recommended changes to the kernel parameters and the security and limits files. For more information, see Chapter 4, “Setting up SUSE Linux Enterprise Server 11 SP2 and Red Hat Enterprise Linux 6.2” on page 55. They are part of the Oracle customization of Linux.

A.6.1 Kernel updates

Make the following updates to the kernel:

```
# echo "kernel.shmmni=4096" >>/etc/sysctl.conf
# echo "kernel.sem=250 32000 100 128" >>/etc/sysctl.conf
# echo "fs.file-max=65536" >>/etc/sysctl.conf
# echo "net.ipv4.ip_local_port_range=1024 65000" >>/etc/sysctl.conf
# echo "net.core.rmem_default=1048576" >>/etc/sysctl.conf
# echo "net.core.rmem_max=1048576" >>/etc/sysctl.conf
# echo "net.core.wmem_default=262144" >>/etc/sysctl.conf
# echo "net.core.wmem_max=262144" >>/etc/sysctl.conf
```

To affect these changes, run the following command:

```
# sysclt -p
```

A.6.2 Changes for pam.d/login

Make the following changes to `pam.d/login`:

```
# echo "session    optional     pam_keyinit.so force revoke" >>/etc/pam.d/login
# echo "session    required     /lib/security/pam_limits.so" >>/etc/pam.d/login
# echo "session    required     pam_limits.so" >>/etc/pam.d/login
```

A.6.3 Changes for limits.conf

Make the following changes to `limits.conf`:

```
# echo "oracle soft nproc 2047" >>/etc/security/limits.conf
# echo "oracle hard nproc 16384" >>/etc/security/limits.conf
# echo "oracle soft nofile 1024" >>/etc/security/limits.conf
# echo "oracle hard nofile 65536" >>/etc/security/limits.conf
```

A.6.4 Setting up Oracle UserID and directories

The Oracle UserID, group, directories, and their ownership must be set up as shown in the following example:

```
# groupadd -g 502 oinstall
# groupadd -g 501  dba
# useradd -u  501 -g oinstall -G dba oracle -m
# passwd oracle
# mkdir /oracle
# chown oracle:dba /oracle
# chmod 755 /oracle
# mkdir /oracle/oracledb
# chown oracle:oinstall /oracle/oracledb
# chmod 755 /oracle/oracledb
# mkdir /oradata
# chown oracle:dba /oradata
# chmod 755 /oradata
```

A.6.5 Disabling SE Linux

The last step in preparation for the installation of the Oracle Database is to disable `SELinux`. To disable this, update `/etc/selinux/config` to reflect `SELINUX=`**`disabled`**.

A.7 Summary

You now have a Red Hat Enterprise Linux 6.3 guest ready for the installation of an Oracle Database 11gR2 (11.2.0.3).

You should review Chapter 4, “Setting up SUSE Linux Enterprise Server 11 SP2 and Red Hat Enterprise Linux 6.2” on page 55 for customization information. Then, review Appendix B, “Installing Oracle and creating a database 11.2.0.3 on Red Hat Enterprise Linux 6” on page 335 as a guide to install an Oracle Database with the My Oracle Support notes and the Oracle installation manuals.

B

Installing Oracle and creating a database 11.2.0.3 on Red Hat Enterprise Linux 6

Attention: Linux commands that are running as non-root are prefixed with `$` in this appendix.

This appendix describes the steps to install Oracle and create a single instance database. This task follows the installation of a Linux guest with Red Hat Enterprise Linux 6 (RHEL 6), as described in Appendix A, “Setting up Red Hat Enterprise Linux 6.3 for Oracle” on page 311 and completing the Linux customization for RHEL 6, as described in 4.2, “Installing Oracle 11.2.0.3 on a Red Hat Enterprise Linux 6 guest” on page 58.

The following overall steps are taken after the Linux setup:

1. Obtain the Oracle code
2. Install the Oracle code
3. Install the latest patch set update
4. Create a database

Not all of the panels are included here. A detailed installation is described in the Oracle documentation and the Red Hat documentation.

This appendix includes the following topics:

- Obtaining the Oracle code and documentation
- Installing the Oracle code
- Upgrading to the latest patch set update level
- Creating an Oracle database

B.1 Obtaining the Oracle code and documentation

Download Oracle 11.2.0.3 for Linux on System z from this website:

http://support.oracle.com

You need the following files:

- `p10404530_112030_Linux-zSer_1of6.zip 1.4 GB`
- `p10404530_112030_Linux-zSer_2of6.zip 1.0 GB`

Oracle Database 11.2.0.3 is a complete replacement for the 11.2.0.2 version, so you do not need to perform an upgrade for a fresh installation.

It is recommended that you install the latest quarterly patch set update (the 11.2.0.3.4 patch set at the time of this writing). Patch 14275605: DATABASE PATCH SET UPDATE 11.2.0.3.4 (INCLUDES CPUOCT2012) was made available October 16th, 2012.

The Oracle 11.2.0.3 Release note and Installation Guide is available at this website:

http://www.oracle.com/pls/db112/homepage

B.2 Installing the Oracle code

The first step is to install 11.2.0.3. When you install 11.2.0.3 on RHEL 6, the OUI fails because it does not pass the prerequisite test. If you start the OUI with the `ignoreSysPrereqs` option, you can proceed, as shown in Figure B-1. Use the following command:

`./runInstaller -ignoreSysPrereqs`

```
[oracle@pazxxt04 ~]$ echo $ORACLE_BASE
/u01/app/
[oracle@pazxxt04 ~]$ cd /scratch/Products/oracle/database/
[oracle@pazxxt04 database]$ ./runInstaller -ignoreSysPrereqs
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 80 MB.  Actual 2840 MB   Passed
Checking swap space: must be greater than 150 MB.  Actual 2047 MB   Passed
Preparing to launch Oracle Universal Installer from
/tmp/OraInstall2012-10-31_08-19-13 AM. Please wait ... [oracle@pazxxt04
database]$
```

Figure B-1 Running runInstaller to install Oracle binaries.

The steps of the installation are shown in the following figures.

Step 1, Register for Security Updates, is shown in Figure B-2.

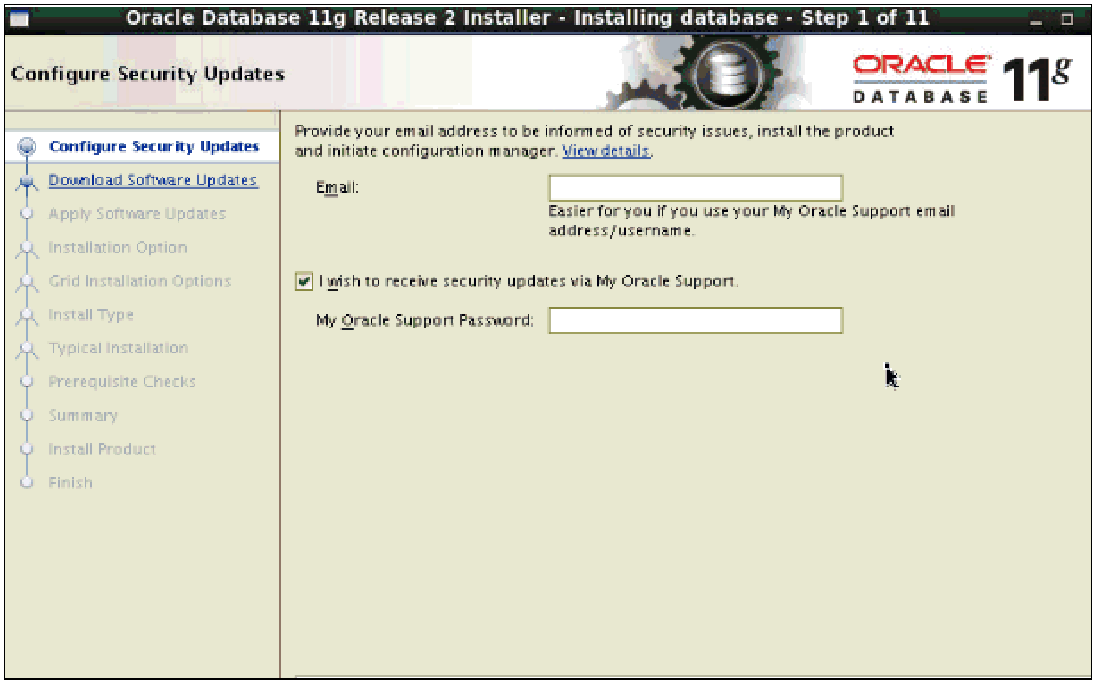


Figure B-2 Step 1: Register for Security Updates

The panels that are shown in Figure B-3 are new with 11gR2.

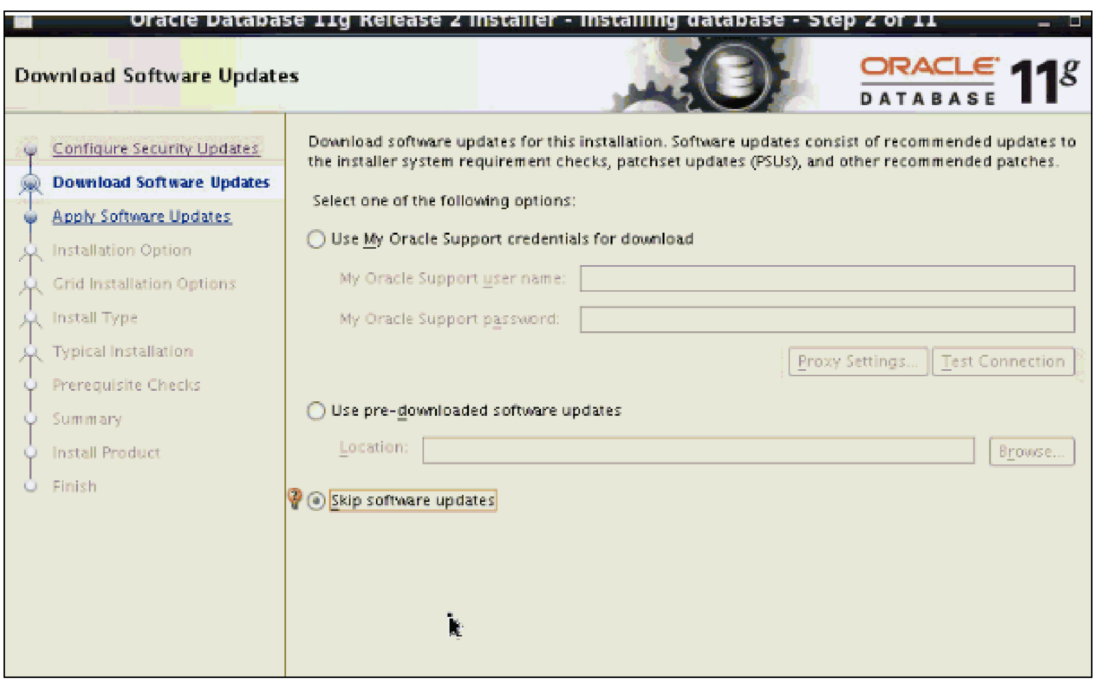


Figure B-3 Step 2: Register for Software Updates

An error message that you receive is shown in Figure B-4.

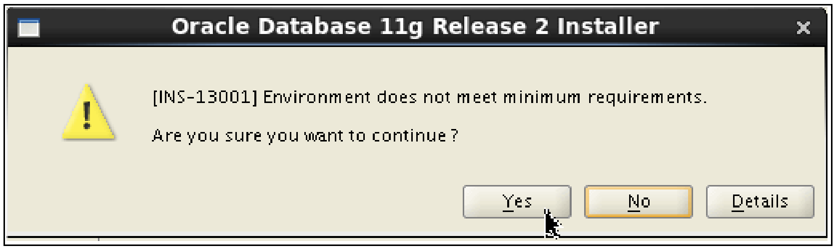


Figure B-4 Error message you receive for RHEL 6 as it is a new version

Because RHEL 6 is a new version of Linux, the verification file is not included and you see an error in the `install_log` file.

Select the type of installation you want, as shown in Figure B-5. In our case, we selected to install the software only for a single instance database.

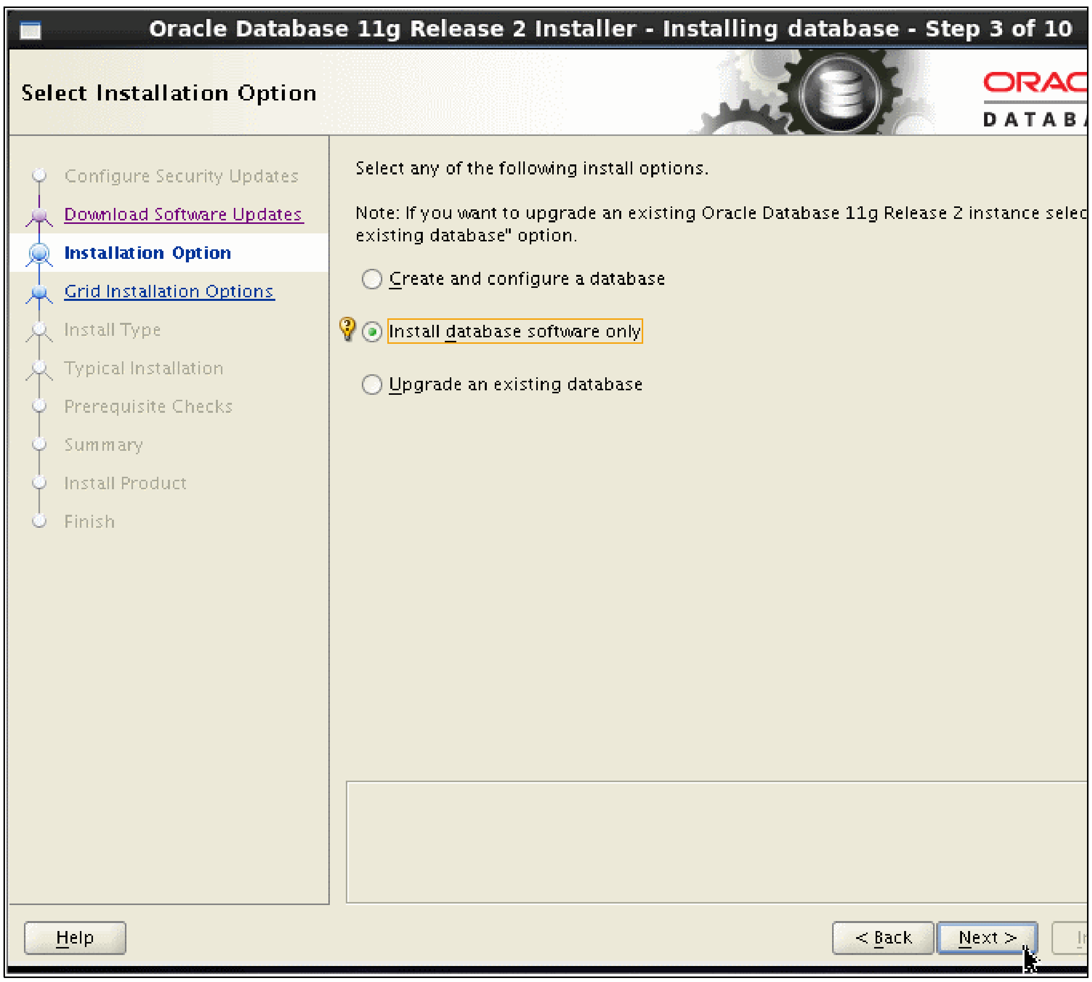


Figure B-5 Step 3: Selecting the type of installation

Step 4 of the process is shown in Figure B-6.

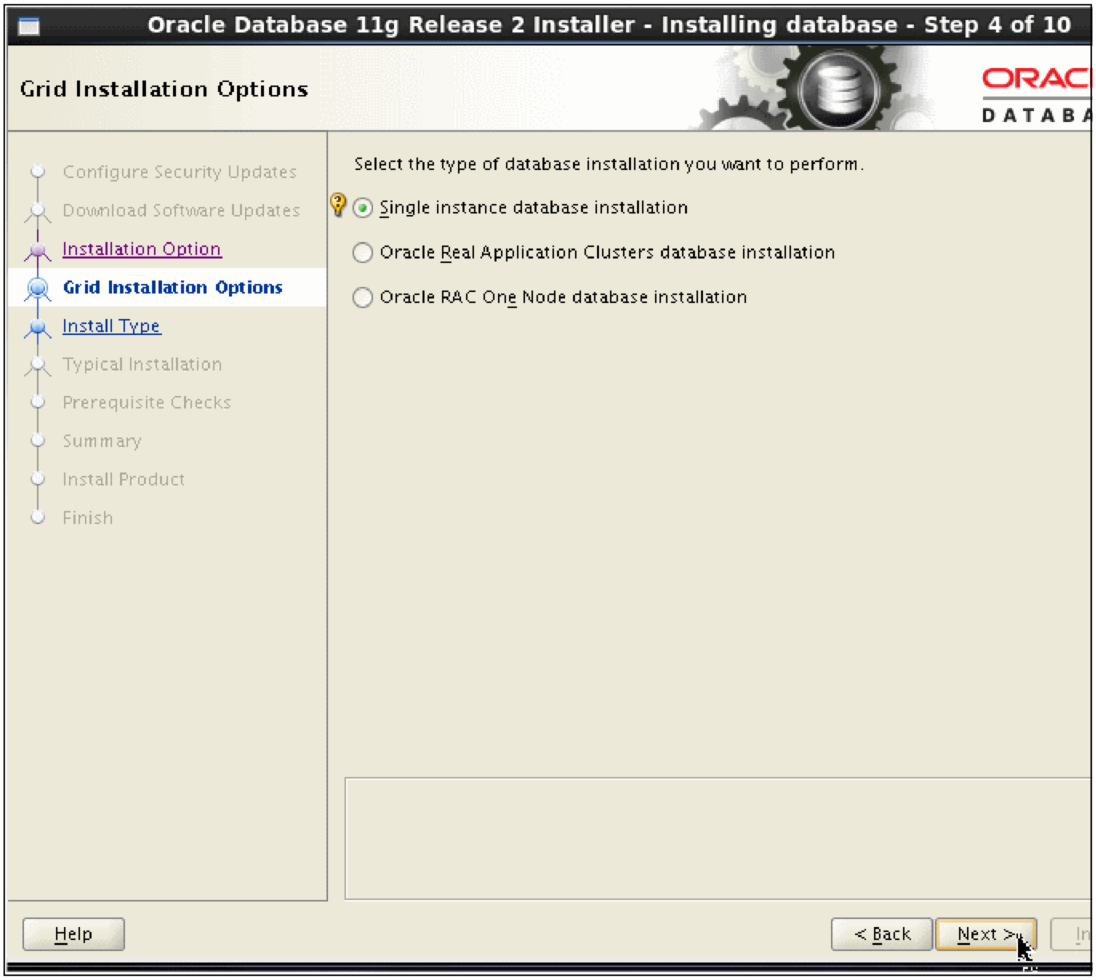


Figure B-6 Step 4: Selecting the Grid Installation Options

Step 5 is shown in Figure B-7. We choose English.

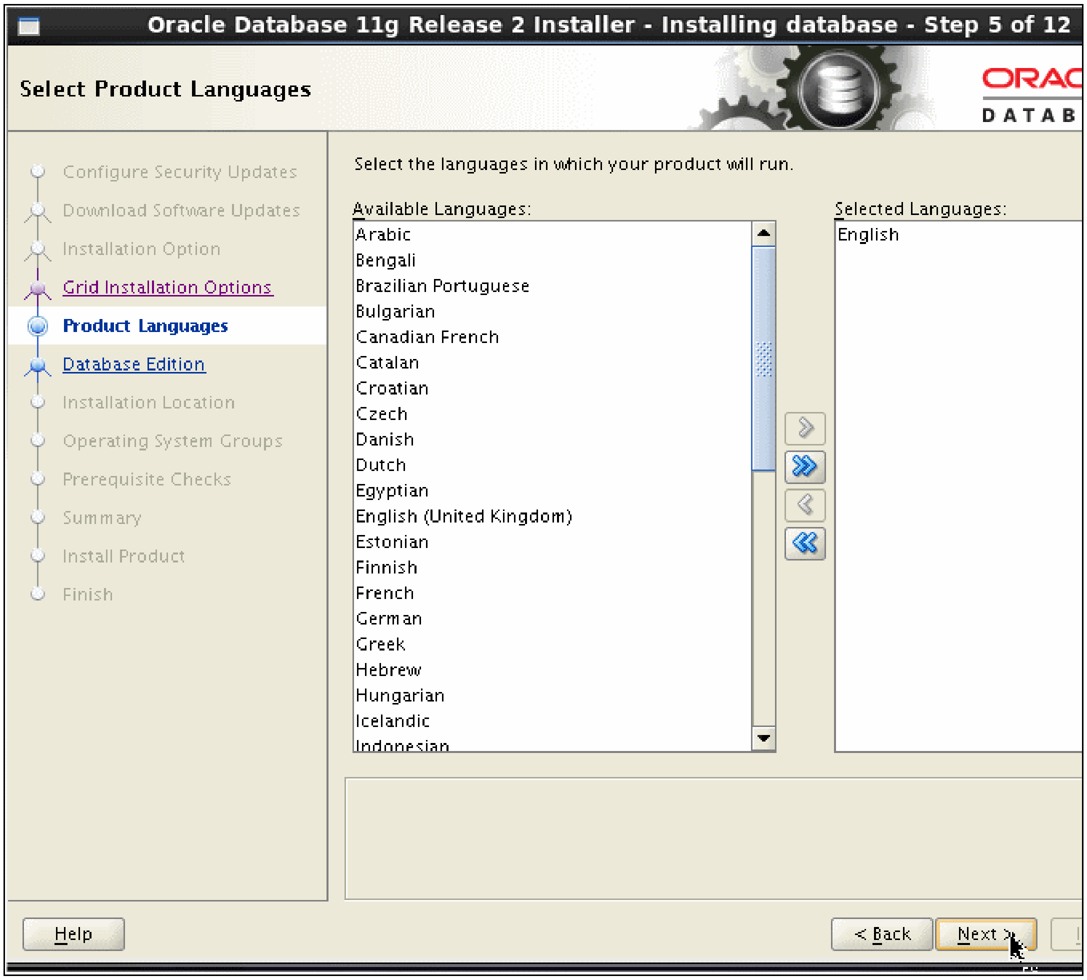


Figure B-7 Step 5: Choose the Language

As shown in Figure B-8, we choose the Enterprise Edition.

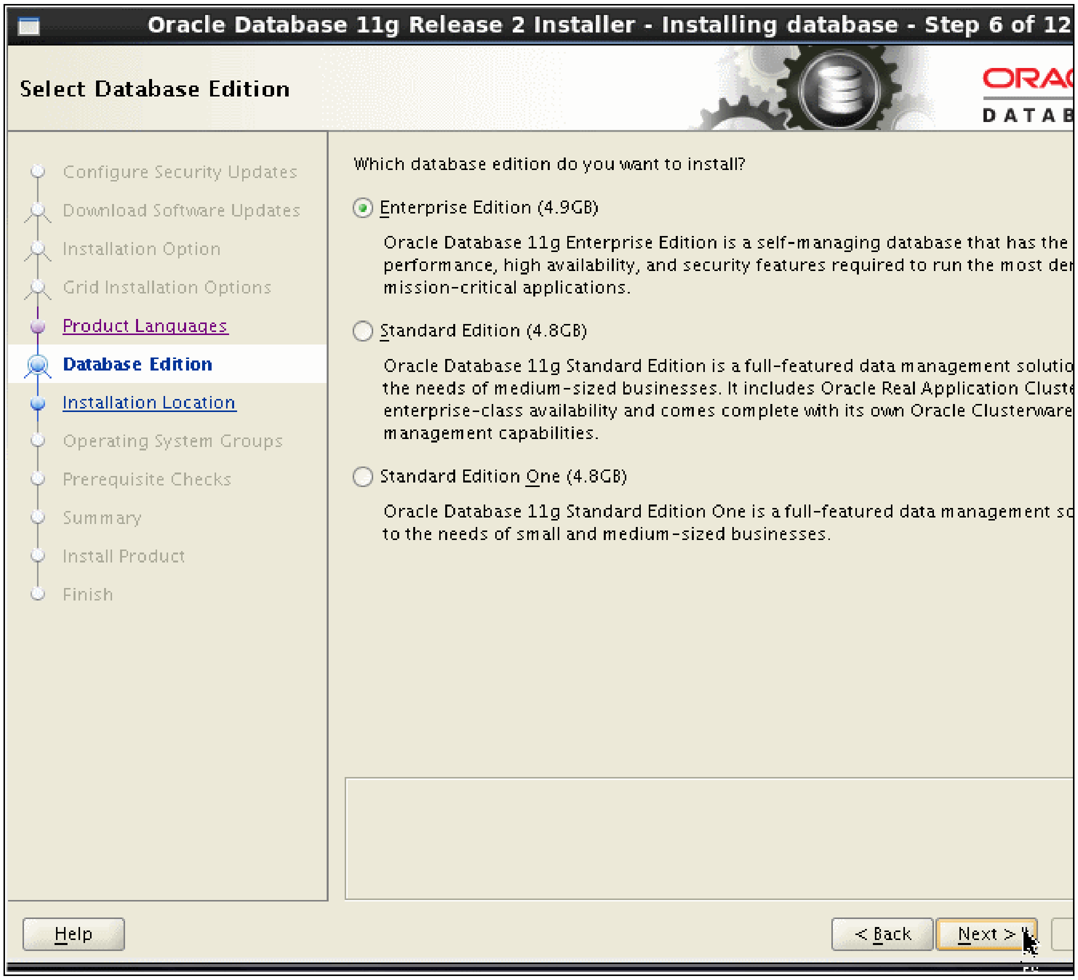


Figure B-8 Step 6: Choose the database edition

As shown in Figure B-9, we specify the location for the installation.

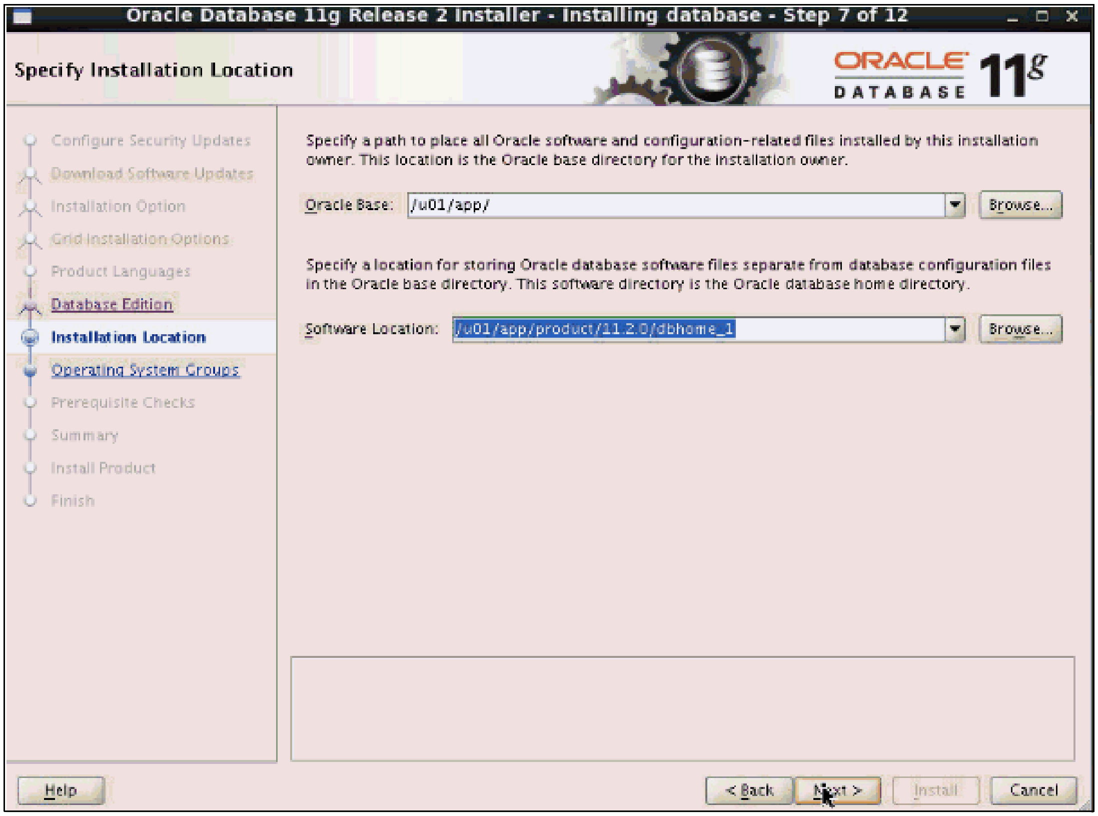


Figure B-9 Step 7: Specify the location for the installation

We specify the location for the inventory, as shown in Figure B-10.

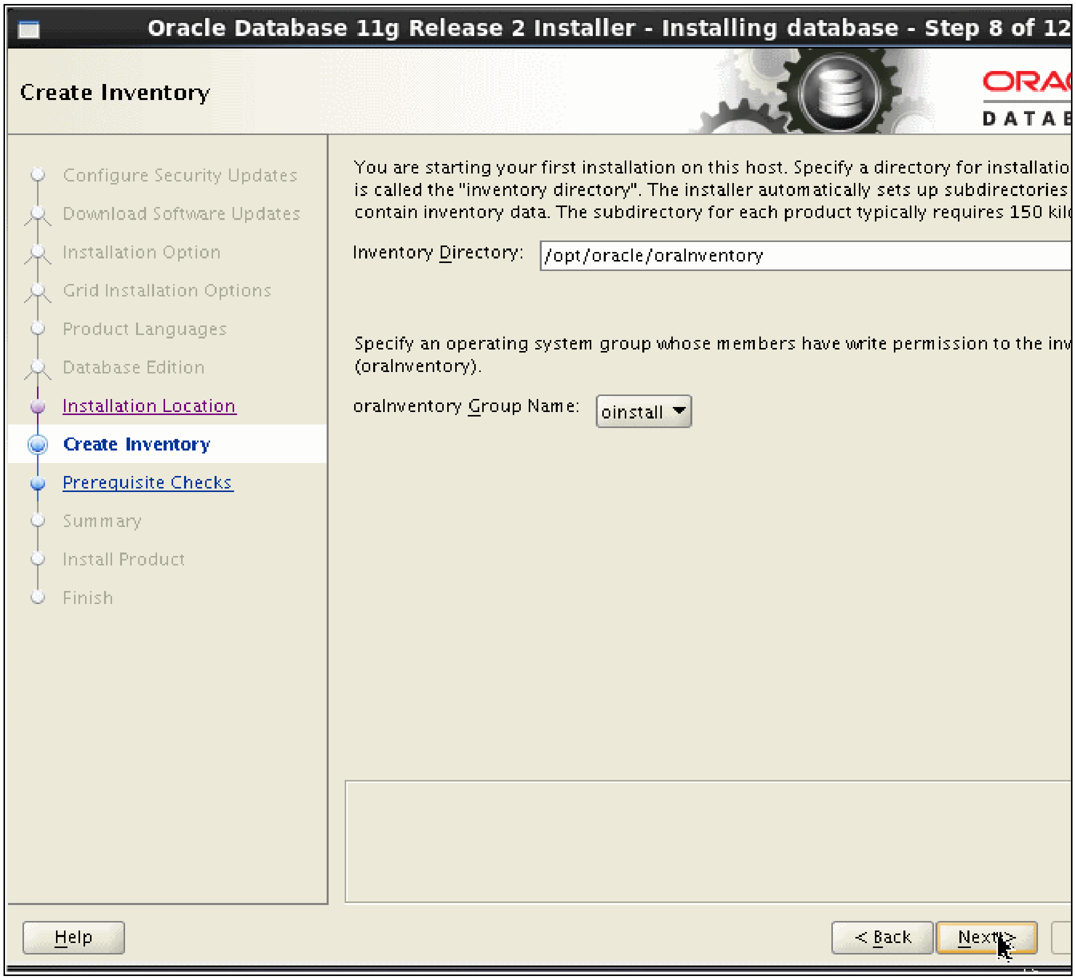


Figure B-10 Location for the inventory

We choose the Group ID, as shown in Figure B-11. We used group `dba`.

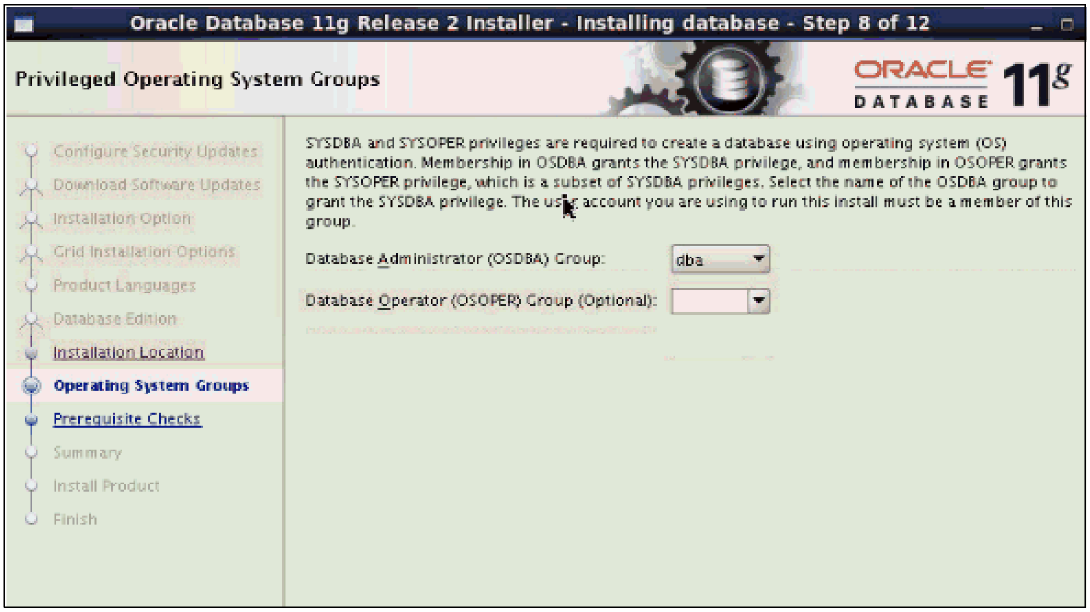


Figure B-11 Step 8: Choose the Group ID

If you entered `ignoreSysprequisites`, the process skips Step 9.

As shown in Figure B-12, a prerequisite check is performed.

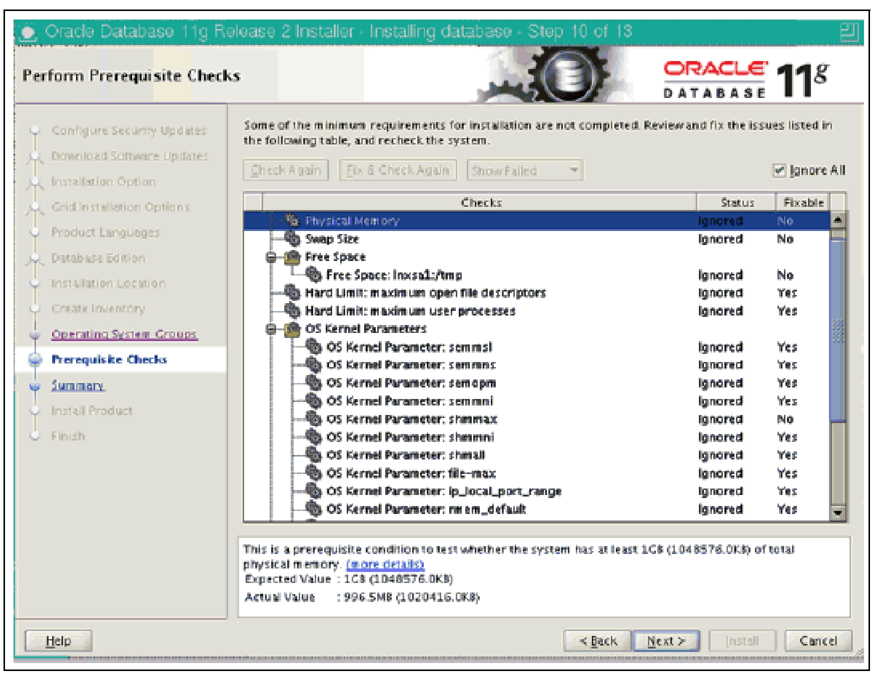


Figure B-12 Prerequisite check

A summary of your choices is shown, as shown in Figure B-13.

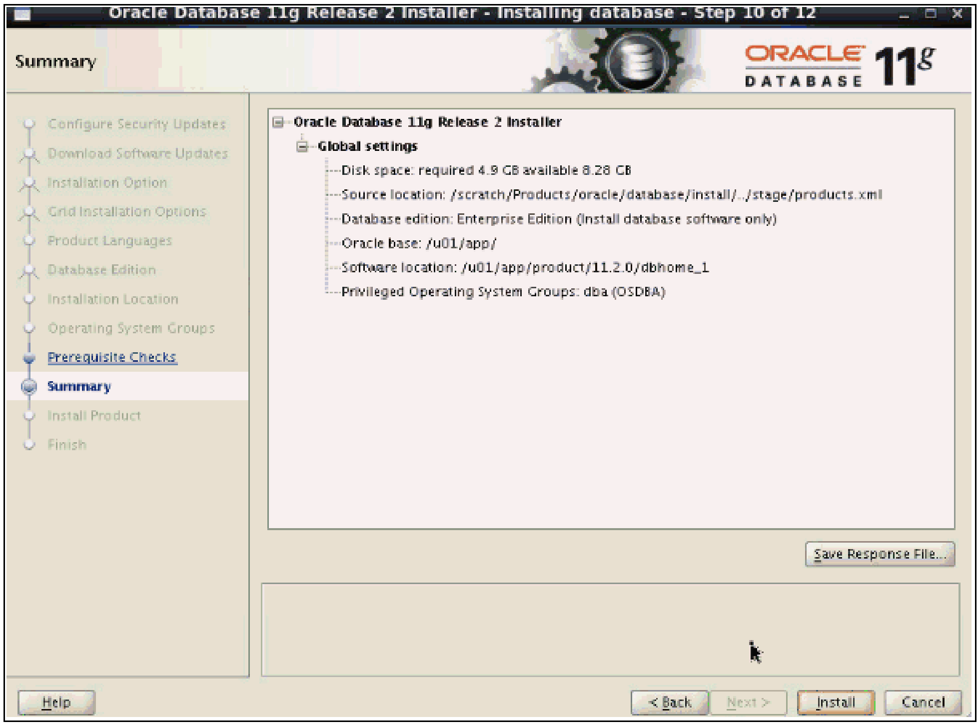


Figure B-13 Step 9 Summary of choices

As shown in Figure B-14, the Oracle binaries are installed.

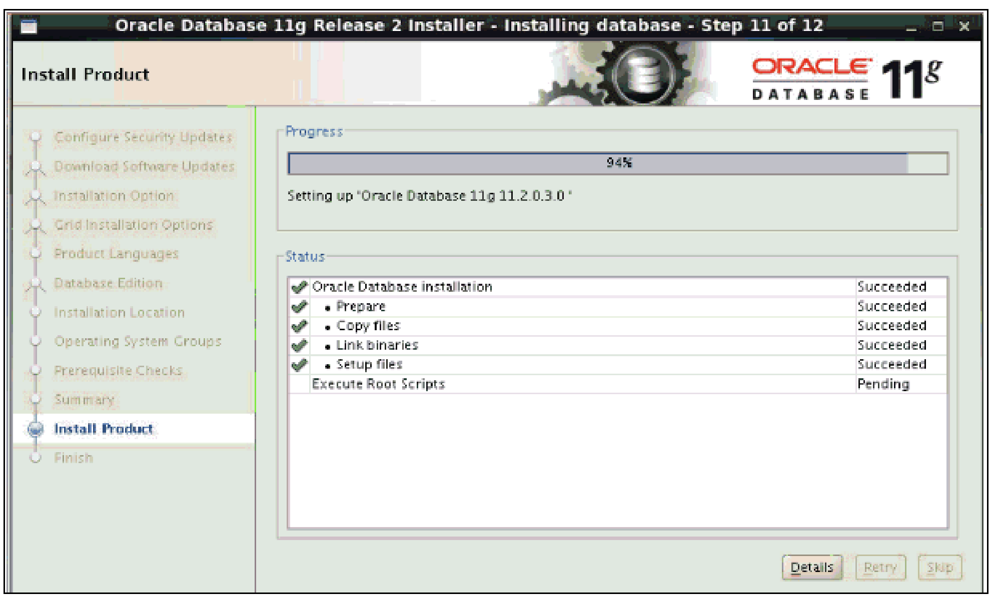


Figure B-14 Step 11: Installing the Oracle binaries

A request run `root.sh` is shown in Figure B-15.

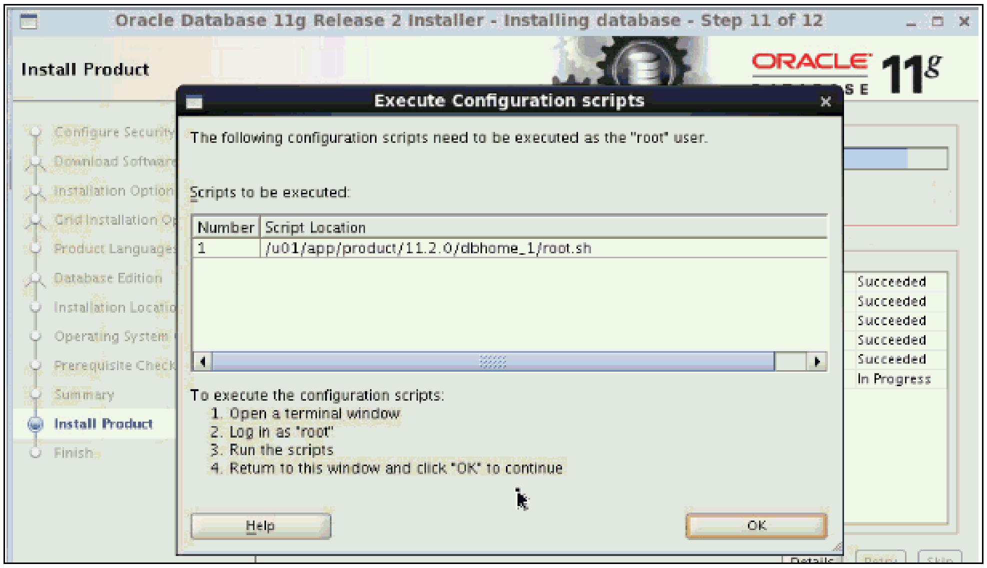


Figure B-15 Request to run root.sh

Figure B-16 shows the result of running `root.sh`.

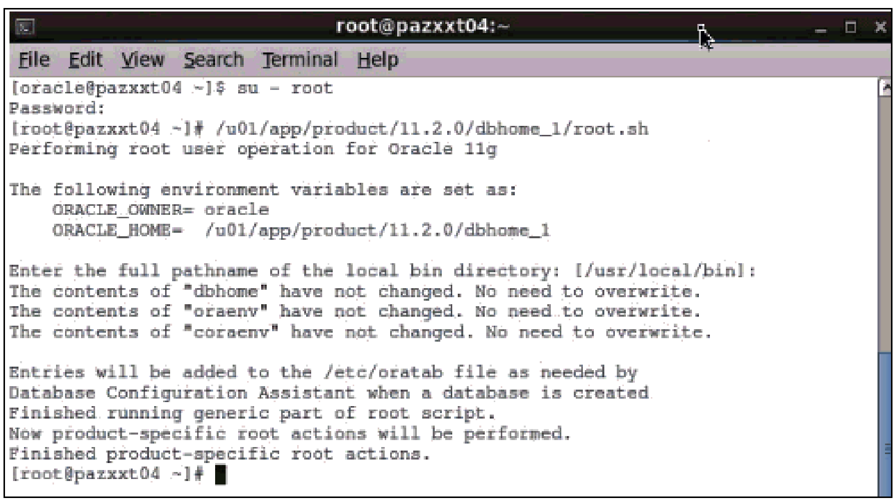


Figure B-16 Run root.sh

The 11.2.0.3 installation runs to completion, as shown in Figure B-17.

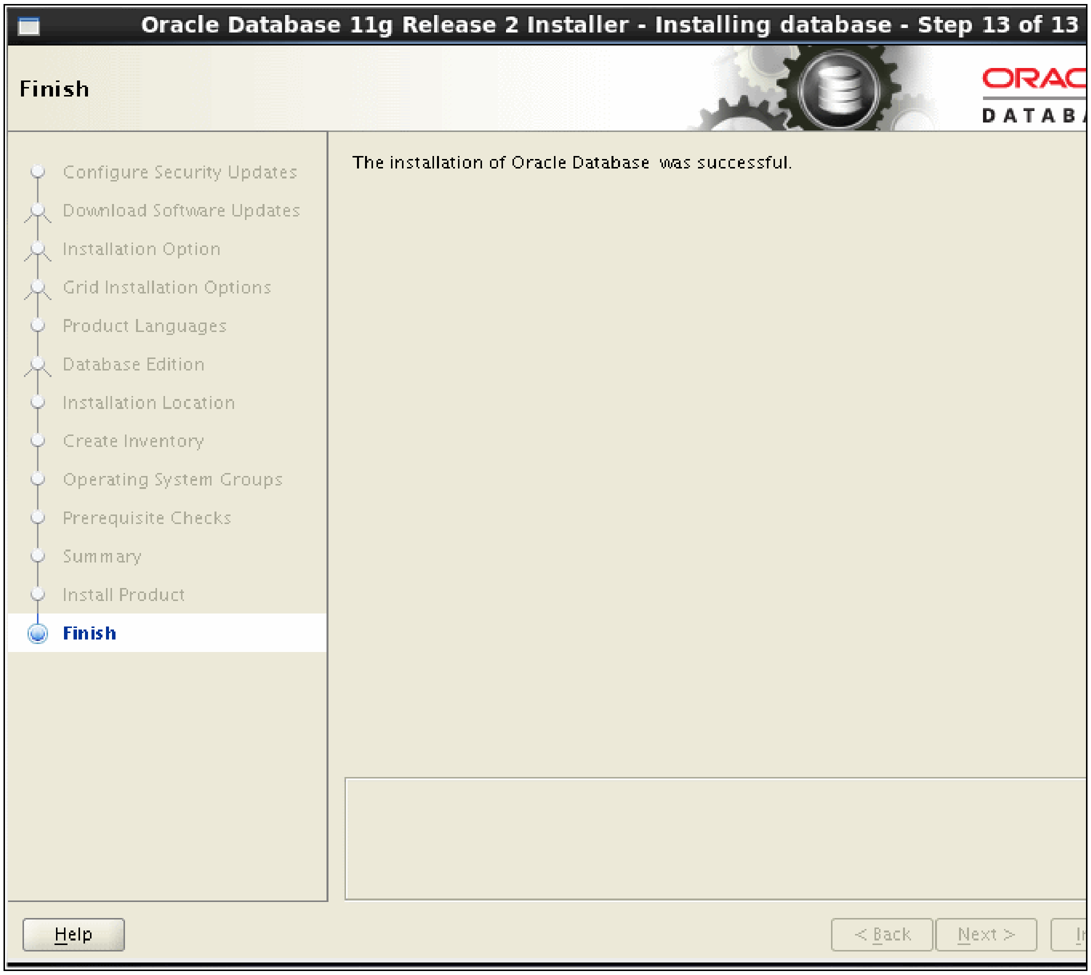


Figure B-17 Step 12: Installation complete

B.3 Upgrading to the latest patch set update level

As a best practice, the next step should be to install the latest patch set update to ensure that the Oracle Database code is the current version.

As of 4Q 2102, the latest DATABASE PATCH SET UPDATE is 11.2.0.3.4, which includes CPUOCT2012. The patch set is 14275605.

The next step is to use DBCA to create a 11.2.0.3.4 database.

B.4 Creating an Oracle database

After 11.2.0.3.4 is installed, you can create a database by using DBCA by running the following command:

```
$ cd $ORACLE_HOME/bin
$ ./dbca
```

After steps in the DBCA installation process are completed, Figure B-18 shows that the database was successfully created.

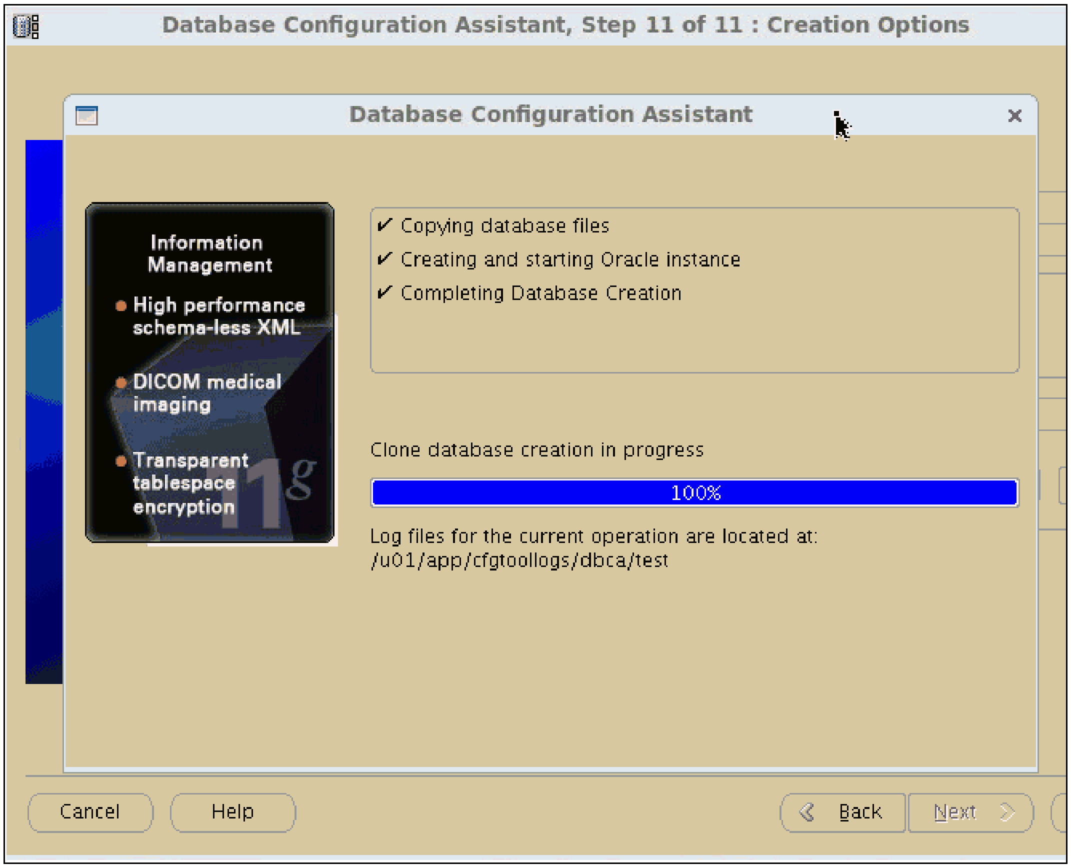


Figure B-18 Database Configuration Assistant

Oracle Database 11gR2 code was installed and a single instance database was created.

C

Working effectively with Oracle support

This appendix is intended to provide an overview of methods and techniques by which you can make effective use of the Oracle Support resources that are provided under their support contracts, in normal operations and when problems are encountered.

We draw extensively on real-world experience (one of the authors is an Oracle Support Engineer) and we hope that it provides a smooth interaction between your organization and Oracle Support.

This appendix includes the following topics:

- Oracle Support for Linux on System z
- Oracle patching process
- Prior planning and preparation prevent poor performance
- Oracle SRs
- Tools

C.1 Oracle Support for Linux on System z

The Oracle database that is provided on Linux on System z is the same Oracle as anywhere else; we take the universal source code and then compile, link, and test it in the new environment. The resulting binaries have different kernel modules included to perform system level operations, as it is a different kernel, but in principle Oracle 11.2 on Red Hat Enterprise Linux x86-64 is the same as Oracle 11.2 on Red Hat Enterprise Linux s390x to a developer or DBA.

We do not have a differential between Linux distributions; it is the same Oracle ISO image for any Linux on System z. We support SUSE Linux Enterprise Server and Red Hat Enterprise Linux in the same way and the distribution is only important for a problem that involves a specific operating system-dependent feature, not for operation of the database.

These two basic tenets mean that most of the functionality of the database and its associated components (for example, RAC and CRS) are identical across platforms, can be operated by a programmer or DBA who is familiar with Oracle on any UNIX version, and, most importantly, can be supported by any Oracle Database Support Engineer. We call these features Generic because they are common to all platforms. For example, common features include all SQL, DDL, and PL/SQL, which is everything you do when you are using database commands and operations.

Because of this commonality, Oracle Support provides teams that are based on functional competencies and are available in many countries and in all time zones to provide a comprehensive, worldwide, 24x7 Support capability. The generic teams that are used include the following teams:

- Database or SQL Performance
- Real Application Clusters and associated technologies
- Automatic Storage Management
- Database Corruption
- Backup and Recovery
- Networking
- Enterprise Manager and Grid Control
- E-Business Suites
- Application servers

That is, there is a team for every product that runs on or interfaces with a database on any platform.

These teams can handle any service request (SR) that does not have platform-specific dependencies. Because it is the same Oracle code that is running the same underlying operations, the platform is irrelevant, and thus there are thousands of support staff members that are trained to handle 99.9% or more of the issues you might encounter in your operation. In our experience, platform-specific issue (those that occur only on, or are caused by, a specific problem in the Linux on System z Architecture) are rare.

However, in addition to the generic Support organization, Oracle provides a team of dedicated Linux on System z engineers, not because we expect platform-specific issues, but because there are some features of the platform, which require skilled, knowledgeable engineers to assist at time. An example of this is z/VM, which is not something that is encountered on any other platform, and an SR that involves interaction between z/VM, Linux and Oracle, and Performance, for example, is greatly assisted by these technical specialists.

This team does not work independently, but with the generic teams to provide specific relevant technical expertise and customer knowledge to a broader generic issue. For example, the following issues require such knowledge:

- Installation and Performance: It helps to know that the Java JDK is provided by IBM, not by Oracle Sun as on other platforms.
- Clustering: Specific knowledge of z/VM Virtual Switch or Hipersocket architecture might be required.

If you have an SR that you believe could benefit from this team's input, you can request in the SR that the Linux on System z team are engaged by including the following text: "Please request the assistance of the Database Specialized Mainframe/Linux on System z team."

When you are opening a new SR, if you choose the following specific information, the SR should be routed directly to the specialist team:

- The correct platform: IBM System z with Linux
- Problem Type: Issues on Linux on System z
- Problem Clarification:
  - Database installation issues on Linux on System z
  - General issues about the database that is running on Linux on System z
  - Performance issues about the database that is running on Linux on System z
  - RAC on Linux on System z

C.2 Oracle patching process

Because this is an area that causes a certain amount of confusion, more information about how this process works might help clarify this area.

All software suffers from defects or bugs, and Oracle is no exception. These are discovered during Oracle's QA process, which permits them to be resolved before the code is released. Unfortunately, no QA process can replicate the enormous variation of customer environments, workloads, and use models, so defects are also found in customer environments.

Oracle diagnosis and patch provision are directly driven by the severity of the customer issue (the technical and business impacts of the defect on customer operations), all other environmental factors are ignored, and platform is not considered. After the defect is diagnosed, a solution is provided, which might not be a patch or code fix. Instead, it might be that a parameter change is an effective solution.

As we stated earlier in this appendix, 99.9% of database operations are generic, thus most of defects are platform independent and can occur anywhere. Most frequently, they were already found and patched. In these cases, the activity is to make the patch available to Linux on System z.

Oracle consists of many millions of lines of C code, which is identical for all platforms (we have the same source tree everywhere). Patching is the process of updating the common source. A patch transaction that affects one or more modules that must then be compiled as object modules for the target platform. These object modules are packaged with metadata for the Oracle Patching Facility (Opatch), and this collection is the basic patch package that is available on My Oracle Support.

The source change is done at one of two levels: the current development level (12c) or the current maintenance level (11.2.0.n). This change is made on the current reference development platform, which changes periodically (currently, Oracle Enterprise Linux). A determination is made whether the change can be merged into older versions (back-level versions). If so, we generate the information to back port the change.

This back porting process is performed in response to customer request, that is, in response to a service request. If we identify that a customer issue is resolved by a particular bug or patch, we can request that the object code is generated for this bug on the customer's platform and Oracle release combination. This generates the Opatch package and loads it to My Oracle Support for a customer to download and apply. A patch of this nature is known as an *interim patch*.

Sometimes, there might be a conflict between one patch that was applied and a new one to be applied. Opatch can detect this and Support usually asks for an Opatch IS inventory, a detailed listing that shows patches that are already applied. Support can then detect potential conflicts before patches are generated.

Conflicts occur because both patches contain changes to the same module, so we must merge these changes before a new module is produced. This process takes longer to perform because there is manual code change and Quality Assurance tests to perform; however, after this is complete, the porting process continues as before.

Because the Interim Patch process is cumbersome and the potential for conflicts can cause delay in providing a solution, we now strongly recommend that customers make full use of the patch collections we provide: *Patch Sets* and *Patch Set Updates*.

Patch Sets are major collections of patches and source updates that generally appear annually and contain solutions for most of serious issues that are discovered since the previous patch set was delivered. The work that is required to integrate the hundreds of bug fixes is enormous. For that reason, there is no specific time frame or schedule for delivery on every platform because quality takes precedence over hard shipping times.

This might sound like a problem, but it was addressed by the second collection type: the Patch Set Update. This consists of solutions for the most serious issues that are discovered that are related to security, integrity, and availability. These are delivered quarterly on a specific schedule and are simultaneously on all platforms, including Linux on System z. The only exception to this rule is if a patch set is shipped close to the scheduled date of a Patch Set Update. This work cannot begin on the update until the Patch Set is available, so in this case, the update is delayed by the time that is required to perform this work. However, this is a rare occurrence.

Thus, by placing Patch Set Updates into their normal maintenance cycle, customers can continually keep their systems at the highest available service level, which proactively avoids many potential issues, and thus maintains service quality standards.

C.3 Prior planning and preparation prevent poor performance

When complex systems with high workloads and even higher business expectations are run, problems must be managed effectively and efficiently to ensure a swift return to normal service. Key to this is the ability to swiftly gather information about the problem, engage the appropriate resources to help with diagnosis and resolution, and then apply the appropriate solution.

C.3.1 Gathering information

There are a number of tools available, from Oracle and elsewhere, that are useful to have available in case of need. Some of these should be running continually, others periodically, and still others only when appropriate to an issue. The critical point for all of these is that they should be installed before they are needed because trying to raise a change control request that requires senior management approval during an outage is not a scenario most system administrators enjoy.

C.3.2 Engaging the appropriate resources

In addition, time should be taken to become familiar with the tools, their output, and correlations with other tools. Most importantly, there should be full and complete interaction between all the administration and operational areas that are associated with the tools and incident response.

The DBA team, application developers, system administrators for Linux, and system programmers for z/VM or System z should all be familiar with each other, their areas of expertise and the tools, concepts, and terminology each use and the diagnostic and investigative procedures that are available. Only then when an issue occurs can a multi-skilled team work closely together to resolve what is, after all, a common issue to their customer service.

C.4 Oracle SRs

At some point, it might be necessary to engage Oracle Support via am SR. It is critical that the SR is opened at one of the following Severity levels that reflects the severity of the issue to you and your business:

- Severity 4: No loss of service or resources
- Severity 3: Minor loss of service or resources
- Severity 2: Severe loss of service or resources without an acceptable temporary solution
- Severity 1: Complete loss of service or resources and work cannot reasonably continue; the work is considered “mission critical”

One point to consider for a Severity 1 SR is whether it is appropriate to be worked 24x7. There is no problem if it is, but if weekend or overnight work is not authorized, you can work an SR during normal working hours, even a Severity 1. This is frequently useful because the SR remains with a single engineer who is familiar with your issue, rather than transferring every 8 hours to a new engineer.

After the SR is open, the Severity might change depending on the status of the issue; for example, if service was restored but we need to find a root cause, this is more appropriately worked at Severity 2 with a single engineer. If the extent of the problem becomes more widespread or an intermittent outage becomes a permanent loss of service, it is appropriate to raise the Severity.

Important: Do not confuse *Severity* with *Escalation*. Severity reflects the effect on your business, whereas escalation means bringing management attention to your SR and, where appropriate, more resources. This direct, two-way communication with a Manager in Support is from where the next action plan comes. Severity increases are discussed during this communication.

If you are dissatisfied with the progress or response to an SR, escalate rather than change the severity because this gets a manager on the case immediately.

C.4.1 Applying the appropriate solution

The appropriate solution is the one that most swiftly restores the service (which can mean different things to many organizations). It might be a parameter change, an application code change, a patch, a change in resource availability, or even a change in working practice or operational scheduling. The key point to remember is that swift restoration of service is key. Secondary operations, such as, root cause analysis or patching to replace a parameter change, while important, are not usually on the critical path to service restoration. A sense of proportion and priority is important.

C.5 Tools

The following list of tools is not a comprehensive list, but a starter. Some tools already might be installed; others might need to be installed or authenticated:

- z/VM:
  - Performance Toolkit, which is available at this website:
  - http://www.vm.ibm.com/related/perfkit/
  - Velocity Software zVPS, which is available at this website:

    http://www.velocity-software.com/product.html
- Linux:
  - Sysstat, which is available at this website:

    http://sebastien.godard.pagesperso-orange.fr/
  - OSWatcher (for more information, see *OSWatcher Black Box Analyzer User Guide*, Doc ID 461053.1)
  - ksar, which is available at this website:

    http://sourceforge.net/projects/ksar/
- Oracle:
  - AWR or Statspack (for more information, see *Performance Tuning Using Advisors and Manageability Features: AWR, ASH, ADDM and Sql Tuning Advisor*, Doc ID 276103.1)
  - LTOM (for more information, see *LTOM - The On-Board Monitor User Guide*, Doc ID 352363.1)
  - RDA/OSWatcher/ProcWatcher

– For RAC/CRS: Cluster Health Monitor/`diagcollection.pl`/Cluvfy

  These are summarized with the full references in *CRS Diagnostic Data Gathering: A Summary of Common tools and their Usage*, Doc ID 783456.1

– Swingbench: A universal load generator that can run in stand-alone or RAC mode, which provides a repeatable load test with multiple transaction type modeled on real world OLTP systems and is available at this website:

  http://www.dominicgiles.com/swingbench.html

– Real Application Testing: Although this is a chargeable feature, the ability to replay your real-world production workload against a clone environment and actually see and measure the result, which permits tuning and problem avoidance before implementation is of such enormous benefit that it should be considered by anyone who is running large or critical systems.

To work most effectively with Oracle Support, ensure that you have these tools installed (especially AWR and OS Watcher) so that you can rapidly provide data from these tools when you open SRs.

D

Additional material

Attention: Linux commands that are running as root are prefixed with `#` in this appendix.

This book refers to additional material that can be downloaded from the Internet as described in the following sections.

Locating the Web material

The Web material that is associated with this book is available in softcopy on the Internet from the IBM Redbooks Web server, which is available at this website:

ftp://www.redbooks.ibm.com/redbooks/SG248104

Alternatively, you can go to the IBM Redbooks website at:

http://www.ibm.com/redbooks

Select **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG24-8104.

Using the Web material

The additional Web material that accompanies this book includes the following file:

File name	*Description*
`SG248104.tgz`	Code samples in a compressed tar file

System requirements for downloading the Web material

The Web material requires the following system configuration:

- Hard disk space:~22 KB
- Operating System: Red Hat Enterprise Linux (RHEL) 6.2
- Memory: 256 MB

Downloading and extracting the Web material

Create a subdirectory on a Linux or Windows system. For Windows, there are many archive or extraction tools that recognize a compressed `.tar` file. For Linux, copy the `.tar` file to a directory and use the **`tar xvf`** command to extract the following contents:

```
# tar xzvf SG248104.tgz
oracleRedbook-SG248104/
oracleRedbook-SG248104/README.txt
oracleRedbook-SG248104/disclaimer.txt
oracleRedbook-SG248104/vm/
oracleRedbook-SG248104/vm/clone.exec
oracleRedbook-SG248104/linux/
oracleRedbook-SG248104/linux/boot.oracle
oracleRedbook-SG248104/linux/boot.onetime
oracleRedbook-SG248104/linux/database.rsp
```

This creates one directory, `oracleRedbook-SG248104/`, with subdirectories `linux/` for Linux files and `vm/` for the one z/VM file.

The **`CLONE EXEC`** is copied to the `MAINT 191` disk. The Linux files are copied to the golden image, as shown in Table D-1.

Table D-1 Linux files copied to the golden image

File	Location	Description
`boot.onetime`	`/etc/init.d/`	First boot script for setting network values
`boot.oracle`	`/etc/init.d/`	First boot script to prepare for Oracle
`database.rsp`	`/tmp/`	Oracle response file for database silent install

For more information about how to use these files, see Chapter 10, “Automating Oracle on System z” on page 199.

Source code listings

The following source code listings are used in this section:

- “CLONE EXEC” on page 359
- “boot.onetime script” on page 360
- “boot.oracle script” on page 364
- “database.rsp response file” on page 378

CLONE EXEC

The following **CLONE EXEC** attempts to use **FLASHCOPY** then falls back to **DDR** if that does not succeed. This clones a Linux on z/VM, which assumes minidisks 100 (Linux), 101 (Oracle), and 302 (swap):

```
/*+------------------------------------------------------------------+*/
/* EXEC to clone minidisks 100 101 and 302 using FLASHCOPY           */
/*+------------------------------------------------------------------+*/
Parse Arg sourceID targetID .
If sourceID = '' | sourceID = '?' | targetID = '' Then Do
  say 'Syntax is:'
  say 'CLONE sourceID targetID'
  exit 1
End

/* verify that the source ID is logged off */
'CP QUERY' sourceID
If rc <> 45 Then Do
  Say sourceID 'does not exist or is not logged off?'
  exit 2
End

Say 'Are you sure you want to overwrite disks on' targetID '(y/n)?'
Parse upper pull answer .
If answer <> 'Y' then
  exit 3

/* FLASHCOY the 100, 101 and 302 disks from sourceID to targetID */
call copyDisk sourceID '100 1100' targetID '100 2100'
call copyDisk sourceID '101 1101' targetID '101 2101'
call copyDisk sourceID '302 1302' targetID '302 2302'

/* start the target virtual machine */
say "Starting new clone" targetID
'CP XAUTOLOG' targetID
exit

/*+------------------------------------------------------------------+*/
copyDisk:
/* copy a minidisk by linking the source R/O and the target R/W then  */
/*   try FLASHCOPY - if it fails, fall back to DDR                    */
/* Parm 1: source user ID                                             */
/* Parm 2: rdev of the minidisk to copy from R/O                      */
/* Parm 3: temporary rdev of the source disk                          */
/*+------------------------------------------------------------------+*/
Arg sourceID vdev1 vdev2 targetID vdev3 vdev4 .

/* Link source disk read-only then target disk read-write */
'CP LINK' sourceID vdev1 vdev2 'RR'
If rc <> 0 Then Do
  say 'CP LINK' sourceID vdev1 vdev2 'RR failed with' rc
  exit 4
End
'CP LINK' targetID vdev3 vdev4 'MR'
If rc <> 0 Then Do
```

```
  say 'CP LINK' targetID vdev3 vdev4 'MR failed with' rc
  exit 5
End

Say 'Trying FLASHCOPY of' vdev2 'to' vdev4 '...'
'CP FLASHCOPY' vdev2 '0 END' vdev4 '0 END'
If (rc <> 0) Then Do                  /* Fallback to DDR             */
  Say 'FLASHCOPY failed, falling back to DDR ...'
                                      /* Queue up DDR commands       */
  Queue 'SYSPRINT CONS'               /* Don't print to file         */
  Queue 'PROMPTS OFF'                 /* Don't ask 'Are you sure?'   */
  Queue 'IN' vdev2 '3390'             /* Input minidisk              */
  Queue 'OUT' vdev4 '3390'            /* Output minidisk             */
  Queue 'COPY ALL'                    /* Copy all contents           */
  Queue ' '                           /* Empty record ends DDR       */
  'DDR'
  retVal = rc
  End
Else retVal = rc

/* Detach the source and target disks */
'CP DETACH' vdev2
'CP DETACH' vdev4
If retVal <> 0 Then
  Say 'Return value from COPYDISK' source target '=' retVal
```

boot.onetime script

The following **boot.onetime** script sets the IP address and host name for a newly cloned Linux at first boot:

```
#!/bin/bash
#
# /etc/init.d/boot.onetime
#
# chkconfig: 345 01 99
### BEGIN INIT INFO
# Provides:          boot.onetime
# Description:       upon first boot find/modify IP@ + hostname, gen SSH keys
### END INIT INFO
#
# This script requires two RHEL 5 parameter files to exist on the user ID's
# 191 disk: (1) the file RH62GOLD PARM-RH5 - the parameter file of the
# golden image and (2) $userid PARM-RHr6 - parameter file of the clone where
# $userid is the ID of the user that is running the script. It then modifies
# the IP address, Host name and fully qualified domain name in three
# configuration files that contain this info. It also regenerates SSH keys
# and checks the SOFTWARE variable to determine if additional scripts need be
# Copied. Finally it turns itself off via "chkconfig" so it only runs once.
#
# IBM DOES NOT WARRANT OR REPRESENT THAT THE CODE PROVIDED IS COMPLETE
# OR UP-TO-DATE.  IBM DOES NOT WARRANT,  REPRESENT OR IMPLY RELIABILITY,
# SERVICEABILITY OR FUNCTION OF THE CODE.  IBM IS UNDER NO OBLIGATION TO
# UPDATE  CONTENT NOR PROVIDE FURTHER SUPPORT.
# ALL CODE IS PROVIDED "AS IS," WITH NO WARRANTIES OR GUARANTEES WHATSOEVER.
```

```
# IBM EXPRESSLY DISCLAIMS TO THE  FULLEST EXTENT PERMITTED BY LAW ALL EXPRESS,
# IMPLIED,  STATUTORY AND OTHER WARRANTIES, GUARANTEES, OR  REPRESENTATIONS,
# INCLUDING, WITHOUT LIMITATION, THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR
# A PARTICULAR  PURPOSE, AND NON-INFRINGEMENT OF PROPRIETARY AND INTELLECTUAL
# PROPERTY RIGHTS.  YOU UNDERSTAND AND AGREE THAT  YOU USE THESE MATERIALS,
# INFORMATION, PRODUCTS, SOFTWARE, PROGRAMS, AND SERVICES, AT YOUR OWN
# DISCRETION AND  RISK AND THAT YOU WILL BE SOLELY RESPONSIBLE FOR ANY DAMAGES
# THAT MAY RESULT, INCLUDING LOSS OF DATA OR DAMAGE TO YOUR COMPUTER SYSTEM.
# IN NO EVENT WILL IBM BE LIABLE TO ANY PARTY FOR ANY DIRECT, INDIRECT,
# INCIDENTAL, SPECIAL, EXEMPLARY OR  CONSEQUENTIAL DAMAGES OF ANY TYPE
# WHATSOEVER RELATED TO OR ARISING FROM USE OF THE CODE FOUND HEREIN, WITHOUT
# LIMITATION, ANY LOST PROFITS, BUSINESS INTERRUPTION, LOST SAVINGS, LOSS OF
# PROGRAMS OR OTHER DATA, EVEN IF IBM IS EXPRESSLY ADVISED OF THE POSSIBILITY
# OF SUCH DAMAGES. THIS EXCLUSION AND WAIVER OF LIABILITY APPLIES TO ALL
# CAUSES OF ACTION, WHETHER BASED ON CONTRACT, WARRANTY, TORT OR ANY OTHER
# LEGAL THEORIES.
#
#+--------------------------------------------------------------------------+
function msg()
# A wrapper around echo that also sends output to the output file in /tmp
# Args: text to echo
#+--------------------------------------------------------------------------+
 {
  echo "$this: $@"
  echo "$@" >> $outFile
 } # msg()

#+--------------------------------------------------------------------------+
function enable191disk()
# Enable the 191 disk and set the global variable disk191
#+--------------------------------------------------------------------------+
 {
  local devLine
  local diskName

  msg "Enabling the 191 disk"
  chccwdev -e 191 > /dev/null 2>&1
  rc=$?
  if [ $rc != 0 ]; then # unable to enable 191 disk
    echo "$this: Unable to enable 191, rc from chccwdev = $rc"
    exit 1
  fi
  sleep 1       # wait a second for disk to be ready
  udevadm settle

  local devLine=`grep "^0.0.0191" /proc/dasd/devices` # the line with 191
  if [ $? = 0 ]; then
    diskName=`echo $devLine | sed -e 's/.* is //g' | awk '{print $1}'`
    disk191="/dev/$diskName"
  else
    msg "Error: 191 disk not found in /proc/dasd/devices"
    exit 2
  fi
 } # enable191disk()
```

```
#+--------------------------------------------------------------------------+
function initialize()
# set up for customizing new clone
#+--------------------------------------------------------------------------+
 {
  this=`basename $0`
  echo "$this - starting at `date`" > $outFile
  thisID=$(cat /proc/sysinfo | grep "VM00 Name" | awk '{print $3}')
  if [ $thisID = "RH62GOLD" ]; then # don't do anything on this ID
    msg "Warning: on golden image RH62GOLD - exiting"
    exit
  fi
  msg "this userID = $thisID"
  enable191disk
 } # initialize()

#+--------------------------------------------------------------------------+
function findSourceIP()
# Get the source IP address and hostName
# Args: none
#+--------------------------------------------------------------------------+
 {
  sourceConf="$sourceID.$confType"
  cmsfslst -d $disk191 | grep $sourceID | grep $confType > /dev/null
  rc=$?
  if [ $rc != 0 ]; then
    echo "$0: $sourceConf not found on 191 minidisk. Exiting"
    exit 2
  fi
  export local $(cmsfscat -a -d $disk191 $sourceConf)
  # set global variable names escaping any dots (.) in the strings
  sourceName=$(echo "$HOSTNAME" | sed -e 's:\.:\\\.:g')
  sourceHost=${HOSTNAME%%.*}  # Chop domain name off to leave host name
  msg "source host name  = $sourceHost"
  sourceIP=$(echo "$IPADDR" | sed -e 's:\.:\\\.:g')
  msg "source IP address = $sourceIP"
  sourceIP2=$(echo "$IPADDR2" | sed -e 's:\.:\\\.:g')
  msg "source IP address 2 = $sourceIP2"
 } # findSourceIP()

#+--------------------------------------------------------------------------+
function findTargetIP()
# Get my new IP address and hostname
# Args: none
#+--------------------------------------------------------------------------+
 {
  targetParm="$thisID.$confType"
  msg "targetParm = $targetParm"
  cmsfslst -d $disk191 | grep $thisID | grep $confType > /dev/null
  rc=$?
  if [ $rc != 0 ]; then
    echo "$0: $targetParm not found on 191 minidisk. Exiting"
    exit 3
  fi
  export local $(cmsfscat -a -d $disk191 $targetParm)
```

```
  targetName=$HOSTNAME
  targetHost=${HOSTNAME%%.*}  # Chop domain name off to leave host name
  msg "target host name = $targetHost"
  targetIP=$IPADDR
  msg "target IP address = $targetIP"
  targetIP2=$IPADDR2
  msg "target IP address 2 = $targetIP2"
 } # findTargetIP()

#+--------------------------------------------------------------------------+
function modifyIP()
# Modify IP address and host name in /etc/hosts, /etc/sysconfig/network and
#  /etc/sysconfig/network-scripts/ifcfg-eth0
# Args: none
#+--------------------------------------------------------------------------+
 {
  # TODO: this function should also modify, DNS, Gateway, broadcast, etc.
  eth0file="/etc/sysconfig/network-scripts/ifcfg-eth0"
  eth1file="/etc/sysconfig/network-scripts/ifcfg-eth1"
  msg "Modifying network values"
  sed --in-place -e "s/$sourceName/$targetName/g" /etc/hosts
  sed --in-place -e "s/$sourceHost/$targetHost/g" /etc/hosts
  sed --in-place -e "s/$sourceIP/$targetIP/g" /etc/hosts
  sed --in-place -e "s/$sourceHost/$targetHost/g" /etc/sysconfig/network
  sed --in-place -e "s/$sourceIP/$targetIP/g" $eth0file
  sed --in-place -e "s/$sourceIP2/$targetIP2/g" $eth1file
  hostname $targetHost

  # change the hostname in the two Oracle response files
  msg "Modifying values in Oracle response files"
  sed --in-place -e "s/HOSTNAME=xxxx/HOSTNAME=$targetName/g" /tmp/database.rsp
 } # modifyIP()

#+--------------------------------------------------------------------------+
function rmSSHkeys()
# Remove the host SSH keys - when sshd starts they will be recreated
# Args: none
#+--------------------------------------------------------------------------+
 {
  rm /etc/ssh/ssh_host_*
 } # rmSSHkeys()

#+--------------------------------------------------------------------------+
# global variables
disk191=""                      # device file name of the 191 disk
sourceID="RH62GOLD"             # VM user ID where first Linux was installed
confType="CONF-RH6"             # File type of configuration file on 191 disk
outFile="/tmp/boot.onetime.out" # the output file
this=""                         # the name of this command

# main()
if [ "$1" = "start" ]; then # configure the system
  initialize "$@"
  findSourceIP
  findTargetIP
```

```
  modifyIP
  rmSSHkeys
  chkconfig boot.onetime off  # run only once => turn self off
fi
```

boot.oracle script

The following **boot.oracle** script prepares a newly cloned RHEL 6.2 Linux system for an installation of Oracle stand-alone or grid software:

```
#!/bin/bash
#
# boot.oracle   Configure this virtual machine for Oracle standalone or grid
#
# chkconfig:    345 98 2
#
# IBM DOES NOT WARRANT OR REPRESENT THAT THE CODE PROVIDED IS COMPLETE
# OR UP-TO-DATE.  IBM DOES NOT WARRANT,  REPRESENT OR IMPLY RELIABILITY,
# SERVICEABILITY OR FUNCTION OF THE CODE.  IBM IS UNDER NO OBLIGATION TO
# UPDATE  CONTENT NOR PROVIDE FURTHER SUPPORT.
# ALL CODE IS PROVIDED "AS IS," WITH NO WARRANTIES OR GUARANTEES WHATSOEVER.
# IBM EXPRESSLY DISCLAIMS TO THE  FULLEST EXTENT PERMITTED BY LAW ALL EXPRESS,
# IMPLIED,  STATUTORY AND OTHER WARRANTIES, GUARANTEES, OR  REPRESENTATIONS,
# INCLUDING, WITHOUT LIMITATION, THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR
# A PARTICULAR  PURPOSE, AND NON-INFRINGEMENT OF PROPRIETARY AND INTELLECTUAL
# PROPERTY RIGHTS.  YOU UNDERSTAND AND AGREE THAT  YOU USE THESE MATERIALS,
# INFORMATION, PRODUCTS, SOFTWARE, PROGRAMS, AND SERVICES, AT YOUR OWN
# DISCRETION AND  RISK AND THAT YOU WILL BE SOLELY RESPONSIBLE FOR ANY DAMAGES
# THAT MAY RESULT, INCLUDING LOSS OF DATA OR DAMAGE TO YOUR COMPUTER SYSTEM.
# IN NO EVENT WILL IBM BE LIABLE TO ANY PARTY FOR ANY DIRECT, INDIRECT,
# INCIDENTAL, SPECIAL, EXEMPLARY OR  CONSEQUENTIAL DAMAGES OF ANY TYPE
# WHATSOEVER RELATED TO OR ARISING FROM USE OF THE CODE FOUND HEREIN, WITHOUT
# LIMITATION, ANY LOST PROFITS, BUSINESS INTERRUPTION, LOST SAVINGS, LOSS OF
# PROGRAMS OR OTHER DATA, EVEN IF IBM IS EXPRESSLY ADVISED OF THE POSSIBILITY
# OF SUCH DAMAGES. THIS EXCLUSION AND WAIVER OF LIABILITY APPLIES TO ALL
# CAUSES OF ACTION, WHETHER BASED ON CONTRACT, WARRANTY, TORT OR ANY OTHER
# LEGAL THEORIES.
#
#+--------------------------------------------------------------------------+
function msg()
# A wrapper to send messages to console and to the output file
# Args: text to echo
#+--------------------------------------------------------------------------+
 {
  echo "$this: $@"
  echo "$@" >> $outFile
 } # msg()

#+--------------------------------------------------------------------------+
function section()
# A wrapper to send messages for the start of a new section
# Args: text to echo
#+--------------------------------------------------------------------------+
 {
  echo "" >> $outFile
```

```
  echo "---------------------------------------------------------" >> $outFile
  echo "$this: $@"
  echo "$@" >> $outFile
  echo "---------------------------------------------------------" >> $outFile
 } # msg()

#+--------------------------------------------------------------------------+
function enable191disk()
# Enable the 191 disk and set the global variable disk191
# Args: none
#+--------------------------------------------------------------------------+
 {
  local devLine
  local diskName

  # disable and enable the 191 disk to pick up any changes
  chccwdev -d 191 > /dev/null 2>&1
  chccwdev -e 191 > /dev/null 2>&1
  rc=$?
  if [ $rc != 0 ]; then # unable to enable 191 disk
    echo "$this: Unable to enable 191, rc from chccwdev = $rc"
    exit 1
  fi
  sleep 1       # wait a second for disk to be ready
  udevadm settle

  local devLine=`grep "^0.0.0191" /proc/dasd/devices` # the line with 191
  if [ $? = 0 ]; then
    diskName=`echo $devLine | sed -e 's/.* is //g' | awk '{print $1}'`
    disk191="/dev/$diskName"
  else
    msg "Error: 191 disk not found in /proc/dasd/devices"
    exit 2
  fi
 } # enable191disk()

#+--------------------------------------------------------------------------+
function getSoftwareVar()
# Get the value of the SOFTWARE variable in the <userID>.CONF-RH6 file
# If it is either "OracleStandalone" or "OracleGrid" then configure the server
# Args: none
#+--------------------------------------------------------------------------+
 {
  local userID=`cat /proc/sysinfo | grep "VM00 Name" | awk '{print $3}'`
  if [ "$userID" = "$goldenID" ]; then # we're on the golden image - exit
    msg "Warning: running on golden image $goldenID - exiting"
    exit 3
  fi
  parmFile="$userID.$confType"
  section "Searching for SOFTWARE variable in parameter file $parmFile"
  cmsfslst -d $disk191 | grep $userID | grep $confType > /dev/null
  if [ $? != 0 ]; then
    msg "Error: $parmFile not found on 191 minidisk"
    exit 4
  fi
```

```
  export local $(cmsfscat -a -d $disk191 $parmFile)
  msg "SOFTWARE variable = $SOFTWARE"
 } # getSoftwareVar()

#+--------------------------------------------------------------------------+
function initialize()
# basic initialization - set variables this and outFile and start the output
# file
#+--------------------------------------------------------------------------+
 {
  this=`basename $0`
  # save one copy of the output file
  if [ -f $outFile ]; then # file exists - make a backup copy
    cp $outFile $outFile.orig
  fi
  echo "$this - starting at `date`" > $outFile
  enable191disk                 # enable the CMS 191 disk
  getSoftwareVar                # check if this is an Oracle server
 }

#+--------------------------------------------------------------------------+
function exitNow()
# Exit the script now with a code and a message
# Arg 1: Exit code
# Remaining args: Command to issue
#+--------------------------------------------------------------------------+
{
  local exitVal=$1
  shift
  msg="$@"
  msg "Severe error: $msg"
  msg "Exiting with $exitVal"
  chkconfig $this off
  exit $exitVal
} # exitNow()

#+--------------------------------------------------------------------------+
function warnCmd()
# Issue a command, check return code and issue warning if non-zero
# Args: Command to issue
#+--------------------------------------------------------------------------+
 {
  cmd="$@"
  msg "cmd: $cmd"
  $cmd >> $outFile 2>&1
  rc=$?
  if [ "$rc" != 0 ]; then # issue warning
    msg "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    msg "Warning: $cmd returned $rc"
    msg "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
  fi
  return $rc
 } # warnCmd()

#+--------------------------------------------------------------------------+
```

```
function exitCmd()
# Issue a command, check return code and abort if non-zero
# Arg 1: Exit code
# Remaining args: Command to issue
#+--------------------------------------------------------------------------+
 {
  local exitVal=$1
  shift
  cmd="$@"
  msg "running: $cmd"
  $cmd
  rc=$?
  if [ "$rc" != 0 ]; then # issue message and abort
    msg "Severe error: $cmd returned $rc"
    msg "Exiting with $exitVal"
    chkconfig $this off
    exit $exitVal
  fi
  return $rc
 } # exitCmd()

#+--------------------------------------------------------------------------+
function checkForRPMs()
# Check for co-requisite Oracle RPMs
# Arg 1: "ora" or "grid"
#+--------------------------------------------------------------------------+
 {
  local type=$1

  if [ ! -f "$oravalRPM" ]; then # co-req RPM not found => error
    msg "Error: $oravalRPM not found - exiting with 5"
    exit 5
  fi
  if [ "$type" = "grid" -a ! -f "$cvuqdiskRPM" ]; then # error
    msg "Error: $cvuqdiskRPM not found - exiting with 6"
    exit 6
  fi

 } # checkForRPMs()

#+--------------------------------------------------------------------------+
function yumInstallRPMs()
# Install s390 and s390x RPMs for Oracle in the global variable "allRPMs"
# Also install "ksh" which only has a s390x flavor. Finally, install the
# Oracle RPM that tests co-reqs: ora-val-rpm-EL6-DB-11.2.0.3-1.s390x.rpm
# Arg 1: "ora" or "grid"
#+--------------------------------------------------------------------------+
 {
  local type=$1
  local nextRPM

  section "Installing all RPMs with yum"
  for nextRPM in $allRPMs; do # install this RPM
    warnCmd yum -y -q install $nextRPM.s390
    warnCmd yum -y -q install $nextRPM.s390x
```

```
  done

  # install the ksh RPM which only has an s390x flavor
  warnCmd yum -y -q install ksh

  # install the Oracle RPM that tests co-reqs
  section "Installing RPM $oravalRPM to test co-reqs"
  warnCmd rpm -i $oravalRPM

  # for grid servers, also install the cvuqdisk RPM
  if [ "$type" = "grid" ]; then # also install the cvuqdisk RPM
    warnCmd rpm -i $cvuqdiskRPM
  fi
 } # yumInstallRPMs()

#+--------------------------------------------------------------------------+
function createGroup()
# Create a group if it doesn't already exist
# Arg 1: The group name
# Arg 2: The group ID
#+--------------------------------------------------------------------------+
 {
  # set the three required parameters
  local theGroup=$1
  local gid=$2

  # check if the group exists
  grep "^$theGroup" /etc/group > /dev/null 2>&1
  if [ $? = 0 ]; then # group exists => issue warning message
    msg "group $theGroup already exists"
  else # group doesn't exist => create it
    warnCmd groupadd -g $gid $theGroup
  fi
 } # createGroup()

#+--------------------------------------------------------------------------+
function createUser()
# Create one user if it doesn't already exist
# Arg 1: The user name
# Arg 2: The user ID
# Arg 3: The primary group
# Arg 4: Additional groups separated by commas
#+--------------------------------------------------------------------------+
 {
  # set the three required parameters
  local user=$1
  local uid=$2
  local mainGrp=$3
  local suppGrps
  local ksh="/bin/ksh"

  # check if the user exists
  id $user > /dev/null 2>&1
  if [ $? = 0 ]; then # user exists => issue warning message
    msg "user $user already exists"
```

```
  else # user doesn't exist => create it
    if [ ${#4} = 0 ]; then # there are no additional groups
      suppGrps=""
    else # 4th arg is additional groups
      suppGrps="-G $4"
    fi
    warnCmd useradd -u $uid -g $mainGrp $suppGrps -m -d /home/$user -s $ksh
$user
    warnCmd chown $user.$mainGrp /home/$user
  fi
 } # createUser()

#+--------------------------------------------------------------------------+
function createGroupsUsers()
# Set up Oracle users and groups
# Args: none
#+--------------------------------------------------------------------------+
 {
  section "Creating Oracle groups and users"
  createGroup oinstall 501
  createGroup dba      502
  createGroup asmadmin 503
  createUser  grid     501 oinstall asmadmin,dba
  createUser  oracle   502 oinstall asmadmin,dba
 } # createGroupsUsers()

#+--------------------------------------------------------------------------+
function setNTP()
# Modify the /etc/sysconfig/ntpd file for Oracle
# Args: none
#+--------------------------------------------------------------------------+
 {
  local NTPfile="/etc/sysconfig/ntpd"

  section "Modifying file $NTPfile"
  warnCmd /bin/cp $NTPfile /tmp/ntpd.orig
  msg "cmd: sed -i -e 's/-u ntp:ntp/-x -u ntp:ntp/g' $NTPfile"
  sed -i -e 's/-u ntp:ntp/-x -u ntp:ntp/g' $NTPfile
  rc=$?
  if [ "$rc" != 0 ]; then
    msg "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    msg "Warning: sed command to /etc/sysconfig/ntp returned $rc"
    msg "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
  fi
  warnCmd chkconfig ntpd on
 } # setNTP()

#+--------------------------------------------------------------------------+
function setKernelParms()
# Set kernel parameters needed for Oracle
# Arg 1: "ora" or "grid"
#+--------------------------------------------------------------------------+
 {
  local type=$1
```

```
  section "Setting kernel parameters in /etc/sysctl.conf"
  warnCmd /bin/cp /etc/sysctl.conf /tmp/sysctl.conf.orig
  echo "#" >> /etc/sysctl.conf
  echo "# Oracle kernel specific parameters" >> /etc/sysctl.conf
  echo "#" >> /etc/sysctl.conf
  echo "fs.file-max = 6815744" >> /etc/sysctl.conf
  echo "fs.aio-max-nr = 1048576" >> /etc/sysctl.conf
  echo "kernel.shmall = 2097152" >> /etc/sysctl.conf
  echo "kernel.shmmax = 4218210304" >> /etc/sysctl.conf
  echo "kernel.shmmni = 4096" >> /etc/sysctl.conf
  echo "kernel.sem = 250 32000 100 128" >> /etc/sysctl.conf
  echo "net.ipv4.ip_local_port_range = 9000 65500" >> /etc/sysctl.conf
  echo "net.core.rmem_default = 262144" >> /etc/sysctl.conf
  echo "net.core.rmem_max = 4194304" >> /etc/sysctl.conf
  echo "net.core.wmem_default = 262144" >> /etc/sysctl.conf
  echo "net.core.wmem_max = 1048576" >> /etc/sysctl.conf
  echo "kernel.spin_retry = 2000" >> /etc/sysctl.conf
  echo "#vm.nr_hugepages = 4000 # for large SGAs > 10 GB" >> /etc/sysctl.conf

  # run sysctl -p so the settings take effect immediately
  warnCmd "sysctl -p" > /dev/null 2>&1
 } # setKernelParms()

#+--------------------------------------------------------------------------+
function setLimitsDotConf()
# Configure the /etc/security/limits.conf file
# Args: none
#+--------------------------------------------------------------------------+
 {
  local limitsFile="/etc/security/limits.conf"

  section "Setting limits in $limitsFile"
  warnCmd /bin/cp $limitsFile /tmp/limits.conf.orig
  echo "# Settings for Oracle:" >> $limitsFile
  echo "grid    soft  nofile   1024" >> $limitsFile
  echo "grid    hard  nofile  65536" >> $limitsFile
  echo "grid    soft  nproc    2047" >> $limitsFile
  echo "grid    hard  nproc   16384" >> $limitsFile
  echo "#" >> $limitsFile
  echo "oracle  soft  nofile   1024" >> $limitsFile
  echo "oracle  hard  nofile  65536" >> $limitsFile
  echo "oracle  soft  nproc    2047" >> $limitsFile
  echo "oracle  hard  nproc   16384" >> $limitsFile
  echo "#" >> $limitsFile
  echo "# Use memlock for Huge Pages support" >> $limitsFile
  echo "#* soft memlock 3145728" >> $limitsFile
  echo "#* hard memlock 3145728" >> $limitsFile
 } # setLimitsDotConf()

#+--------------------------------------------------------------------------+
function setUlimits()
# Set two ulimits for a user
# Arg 1: "ora" or "grid"
#+--------------------------------------------------------------------------+
 {
```

```
  local profile
  user=$1

  if [ "$user" = "ora" ]; then # use oracle global var
    profile=$oraProfile
  else
    profile=$gridProfile
  fi

  echo 'ulimit -n 65536' >> $profile
  echo 'ulimit -u 16384' >> $profile
 } # setUlimits()

#+--------------------------------------------------------------------------+
function createOraProfile()
# Set environment variables and PATH variable in oracle's .profile
# Args: none
#+--------------------------------------------------------------------------+
 {
  if [ ! -d /home/oracle ]; then
    msg "Error: /home/oracle does not exist"
    exit 7
  fi

  # set environment variables for the oracle user
  echo "export ORACLE_HOME=/opt/oracle/11.2" > $oraProfile
  echo "export ORACLE_BASE=/opt/oracle" >> $oraProfile
  echo "export ORACLE_SID=orcl" >> $oraProfile
  echo 'export PATH=$PATH:$ORACLE_HOME/bin' >> $oraProfile
  chown oracle.oinstall $oraProfile

  # add ulimit commands
  setUlimits oracle
 } # createOraProfile()

#+--------------------------------------------------------------------------+
function createGridProfile()
# Set environment variables and PATH variable in grid's .profile
# Args: none
#+--------------------------------------------------------------------------+
 {
  if [ ! -d /home/grid ]; then
    msg "Error: /home/grid does not exist"
    exit 8
  fi

  # set environment variables for the grid user
  echo "export GRID_HOME=/opt/grid/??" > $gridProfile
  echo "export GRID_BASE=/opt/grid" >> $gridProfile
  echo 'export PATH=$PATH:$GRID_HOME/bin' >> $gridProfile
  chown grid.oinstall $gridProfile

  # add ulimit commands
  setUlimits grid
 } # createGridProfile()
```

```
#+--------------------------------------------------------------------------+
function enableLUNs()
# Create the config file /etc/zfcp.conf and run zfcpconf.sh which onlines LUNs
# Args: none
# return: 0 = success
#         1 = one of 4 FCP variables not set
#+--------------------------------------------------------------------------+
 {
  if [ ${#FCP400WWPN} = 0 ]; then
    msg "Warning: FCP400WWPN is not set in $parmFile"
    return 1
  fi
  if [ ${#FCP500WWPN} = 0 ]; then
    msg "Warning: FCP500WWPN is not set in $parmFile"
    return 1
  fi
  if [ ${#FCPLUN1} = 0 ]; then
    msg "Warning: FCPLUN1 is not set in $parmFile"
    return 1
  fi
  if [ ${#FCPLUN2} = 0 ]; then
    msg "Warning: FCPLUN2 is not set in $parmFile"
    return 1
  fi

  # export the variables in the CONF file and create 4 LUNs in /etc/zfcp.conf
  section "Enabling LUNs in /etc/zfcp.conf with zfcpconf.sh"
  echo "0.0.0400 $FCP400WWPN $FCPLUN1" >> /etc/zfcp.conf
  echo "0.0.0400 $FCP400WWPN $FCPLUN2" >> /etc/zfcp.conf
  echo "0.0.0500 $FCP500WWPN $FCPLUN1" >> /etc/zfcp.conf
  echo "0.0.0500 $FCP500WWPN $FCPLUN2" >> /etc/zfcp.conf

  # run zfcpconf.sh to configure LUNs which reads /etc/zfcp.conf
  msg "cmd: /sbin/zfcpconf.sh"
  /sbin/zfcpconf.sh # can't check rc as it always returns 1
 } # enableLUNs()

#+--------------------------------------------------------------------------+
function mkLogicalVolume()
# Make a logical volume named from the partitions passed in with the
# following characteristics:
#   Number of stripes = num partitions (-i $numPartitions)
#   Stripe size = 64 KB (-I 64)
#   Read ahead = off (-r 0)
#   Use all space (-l 100%VG)
#   LV name = oradata_lv (-n oradata_lv) - set in the dataName global var
# Args: partitions from which to make the logical volume
#+--------------------------------------------------------------------------+
 {
  local allPartitions="$@"
  local numPartitions="$#"

  # make physical volumes of each of the partitions
  section "Creating logical volume for data"
```

```
  for nextPartition in $allPartition; do
    msg "Making partition on $nextPartition"
    exitCmd 9 pvcreate $nextPartition
  done

  # make a volume group named $dataName_vg of the physical volumes
  msg "Creating volume group ${dataName}_vg"
  warnCmd vgcreate ${dataName}_vg $allPartitions

  # make the logical volume
  msg "Creating logical volume ${dataName}_lv"
  LVargs="-i $numPartitions -I 64 -r 0 -l 100%VG -n ${dataName}_lv"
  warnCmd "lvcreate $LVargs ${dataName}_vg"
  warnCmd udevadm settle
 } # mkLogicalVolume()

#+--------------------------------------------------------------------------+
function mountLogicalVolume()
# Mount the logical volume
# Arg 1: logical volume to mount
# Arg 2: mount point
#+--------------------------------------------------------------------------+
 {
  local lv=$1
  local mountPt=$2

  # make the mount point if it doesn't exist
  section "Mounting the logical volume /dev/${dataName}_vg/${dataName}_lv"
  if [ ! -d $mountPt ]; then # mount point does not exist => create it
    warnCmd mkdir $mountPt
  fi

  # add the file system and LV to /etc/fstab if not already there
  grep "^/dev/mapper/${dataName}_vg-${dataName}_lv" /etc/fstab > /dev/null 2>&1
  if [ $? != 0 ]; then # LV name not in fstab => append it
   local fstabLine="/dev/mapper/${dataName}_vg-${dataName}_lv /${dataName}
ext4    defaults        0 0"
    warnCmd cp /etc/fstab /tmp/fstab.orig
    echo "$fstabLine" >> /etc/fstab
  fi

  # make a file system on the logical volume
  msg "Making an ext4 file system on $lv"
  warnCmd mkfs.ext4 $lv

  # make oracle the owner of the data mount point
  msg "Making oracle the owner of $mountPt"
  warnCmd chown oracle.oinstall $mountPt

  # mount the LV over the mount point
  msg "Mounting $lv over $mountPt"
  warnCmd mount $lv $mountPt
 } # mountLogicalVolume()

#+--------------------------------------------------------------------------+
```

```
function setFCPdevices()
# Given two FCP devices (400 and 500) and two LUNs, set the LUNs online,
# make an LVM out of them and mount it over /oradata
# Further assumptions are that there are two LUNs and one WWPN for each
# of the 400 and 500 FCP devices. The values are set in variables:
# FCP<device>WWPN, FCP<device>LUN1 and FCP<device>LUN2
# Arg 1: "ora" or "grid"
# return: 0 = success
#         1 = enableLUNs() failed
#+--------------------------------------------------------------------------+
 {
  local type=$1

  # enable the two LUNs over two channel paths
  enableLUNs
  if [ $? != 0 ]; then # unable to enable LUNs
    msg "Warning: enableLUNs() failed - not setting up LUNs"
    return 2
  fi

  # start the multipathd service
  exitCmd 10 service multipathd start
  warnCmd sleep 1
  warnCmd udevadm settle

  # stop and start multipathing
  msg "stopping multipathd"
  warnCmd service multipathd stop           # stop the service
  msg "Running: multipath -F"               # issue a message
  multipath -F                              # flush the cache
  warnCmd chkconfig multipathd on           # start multipath service at boot
  warnCmd service multipathd start          # start the service now
  warnCmd multipath                         # run multipath one time

  # for Oracle standalone, make a logical volume of the LUNs then mount it
  if [ "$type" = "ora" ]; then # make LV and mount it
    mkLogicalVolume /dev/mapper/mpatha /dev/mapper/mpathb
    mountLogicalVolume /dev/${dataName}_vg/${dataName}_lv /$dataName
  else # voting disks and data FCP LUNs will be controlled by ASM
    setDiskOwnership
  fi
 } # setFCPdevices()

#+--------------------------------------------------------------------------+
function mkDirectories()
# Make HOME and BASE directories for oracle and grid
# Arg 1: "ora" or "grid"
#+--------------------------------------------------------------------------+
 {
  local type=$1

  # make the data directory first
  warnCmd mkdir /$dataName
  warnCmd chown oracle.oinstall /$dataName
```

```
  # make the inventory directory
  warnCmd mkdir /opt/oraInventory
  warnCmd chown oracle.oinstall /opt/oraInventory

  # make the directories for binaries
  warnCmd mkdir $baseDir/oracle
  warnCmd mkdir $baseDir/oracle/11.2
  if [ "$type" = "grid" ]; then # also create grid directories
    warnCmd mkdir /opt/gridInventory
    warnCmd chown grid.oinstall /opt/gridInventory
    warnCmd mkdir $baseDir/grid
    warnCmd mkdir $baseDir/grid/base
    warnCmd mkdir $baseDir/grid/11.2
    warnCmd chown -R grid.oinstall $baseDir/grid
  fi
  warnCmd chown -R oracle.oinstall $baseDir/oracle
  warnCmd chmod -R 775 $baseDir
 } # mkDirectories()

#+--------------------------------------------------------------------------+
function mkResponseFile()
# Copy a response file in /tmp and modify certain settings based on i
#   variables in the CONF file:
#   PASSWORD - A common password to be used (ora and grid)
#   SCANNAME - The scanName and clusterName variables (grid only)
#   NODELIST - the nodeList varaible (grid only)
# Arg 1: "ora" or "grid"
#+--------------------------------------------------------------------------+
 {
  local type=$1

  # make a database.rsp file in the oracle home directory
  cat /tmp/database.rsp | sed \
    -e "s/Password=xxxx/Password=$PASSWORD/g" \
    -e "s/clusterNodes=xxxx/clusterNodes=xxxx$NODELIST/g" \
    -e "s/ALL=xxxx/ALL=$PASSWORD/g" > /home/oracle/database.rsp
  chown oracle.oinstall /home/oracle/database.rsp
 } # mkResponseFile()

#+--------------------------------------------------------------------------+
function setDiskOwnership()
# Add rules the a new file, 99-udev-oracle.rules, for three voting disks
200-202
# Args: none
#+--------------------------------------------------------------------------+
 {
  udevFile=/etc/udev/rules.d/99-udev-oracle.rules
  rcFile=/etc/rc.d/rc.local

  # enable the three voting disks
  for vdev in 200 201 202; do
    warnCmd "chccwdev -e $vdev"
  done

  # set the ownership of the three voting disks with udev rules
```

```
  echo '# change ownership of disks for Oracle ASM' > $udevFile
  echo 'KERNEL=="dasd*1",ID=="0.0.0200",OWNER="grid",GROUP="dba",MODE="0660"'
>> $udevFile
  echo 'KERNEL=="dasd*1",ID=="0.0.0201",OWNER="grid",GROUP="dba",MODE="0660"'
>> $udevFile
  echo 'KERNEL=="dasd*1",ID=="0.0.0202",OWNER="grid",GROUP="dba",MODE="0660"'
>> $udevFile

  # set the ownership of the two data disks with the /etc/init.d/rc.local file
  echo "" >> $rcFile
  echo "# Set ownership of two data disks for Oracle ASM" >> $rcFile
  echo "chown oracle:asmadmin /dev/mapper/mpatha" >> $rcFile
  echo "chown oracle:asmadmin /dev/mapper/mpathb" >> $rcFile
 } # setDiskOwnership()

#+--------------------------------------------------------------------------+
function configureOracle()
# Configure a new Linux system for Oracle either standalone database or grid
#   cluster
# Args: none
#+--------------------------------------------------------------------------+
 {
  if [ "$SOFTWARE" = "OracleStandalone" ]; then    # Oracle standalone server
    msg "Preparing for Oracle standalone installation"
    checkForRPMs ora             # check for Oracle RPMs
    createGroupsUsers ora        # define users and groups
    yumInstallRPMs ora           # add RPMs necessary for Oracle with yum
    setNTP                       # configure the NTP service
    setLimitsDotConf             # configure limits.conf file
    createOraProfile             # set up .profile for oracle user
    setKernelParms ora           # configure kernel parameters
    setFCPdevices ora            # set FCP devices online
    mkDirectories ora            # make directories for Oracle standalone
    mkResponseFile ora           # customize the database response file
  elif [ "$SOFTWARE" = "OracleGrid" ]; then   # Oracle CRS cluster server
    msg "Preparing for Oracle grid installation"
    checkForRPMs grid            # check for Oracle RPMs
    createGroupsUsers grid       # define users and groups
    yumInstallRPMs grid          # add RPMs necessary for Oracle with yum
    setNTP                       # configure the NTP service
    setLimitsDotConf             # configure limits.conf file
    createOraProfile             # set up .profile for the oracle user
    createGridProfile            # set up .profile for the grid user
    setKernelParms grid          # configure kernel parameters
    setFCPdevices grid           # set FCP devices online
    mkDirectories grid           # make directories for Oracle grid
    mkResponseFile grid          # customize the grid response file
  else                           # not an Oracle system
    msg "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    msg "SOFTWARE varaiable not set to 'OracleStandalone' or 'OracleGrid' -
exiting"
    msg "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    exit 11
  fi
 } # configureOracle()
```

```
#+--------------------------------------------------------------------------+
function successMsg()
# Issue a success message
# Args: none
#+--------------------------------------------------------------------------+
 {
  msg "*"
  msg "*********************************************************"
  msg "Sucessfully completed!"
  msg "*********************************************************"
 } # successMsg()

#+--------------------------------------------------------------------------+
# global variables
allRPMs="compat-libcap1 compat-libstdc++-33 elfutils-libelf-devel libaio-devel"
confType="CONF-RH6"                     # File type of configuration file
dataName="oradata"                      # Oracle data mount point, vg name,
etc
baseDir="/opt"                          # directory with LV for Oracle
binaries
disk191=""                              # device file name of the 191 disk
goldenID="RH62GOLD"                     # user ID of the golden image
outFile="/tmp/boot.oracle.out"          # the output file
gridProfile=/home/grid/.profile         # the grid user's profile
oraProfile=/home/oracle/.profile        # the oracle user's profile
cvuqdiskRPM="/tmp/cvuqdisk-1.0.9-1.rpm"  # two required Oracle RPMs
oravalRPM="/tmp/ora-val-rpm-EL6-DB-11.2.0.3-1.s390x.rpm"
parmFile=""                             # the CMS parameter file
this=""                                 # the name of this command

# main()
if [ "$1" = "start" ]; then # configure the system
  initialize                            # set up
  configureOracle                       # do the real work
  chkconfig boot.oracle off             # turn self off so runs just once
  successMsg                            # success if we fall through to here
fi
```

database.rsp response file

The following **database.rsp** response file is used to silently install Oracle database:

```
oracle.install.responseFileVersion=/oracle/install/rspfmt_dbinstall_response_sc
hema_v11_2_0
oracle.install.option=INSTALL_DB_AND_CONFIG
ORACLE_HOSTNAME=xxxx
UNIX_GROUP_NAME=oinstall
INVENTORY_LOCATION=/opt/oraInventory
SELECTED_LANGUAGES=en
ORACLE_HOME=/opt/oracle/11.2
ORACLE_BASE=/opt/oracle
oracle.install.db.InstallEdition=EE
oracle.install.db.EEOptionsSelection=false
oracle.install.db.optionalComponents=oracle.rdbms.partitioning:11.2.0.3.0,oracl
e.oraolap:11.2.0.3.0,oracle.rdbms.dm:11.2.0.3.0,oracle.rdbms.dv:11.2.0.3.0,orac
le.rdbms.lbac:11.2.0.3.0,oracle.rdbms.rat:11.2.0.3.0
oracle.install.db.DBA_GROUP=dba
oracle.install.db.OPER_GROUP=dba
oracle.install.db.isRACOneInstall=false
oracle.install.db.config.starterdb.type=GENERAL_PURPOSE
oracle.install.db.config.starterdb.globalDBName=rac.us.oracle.com
oracle.install.db.config.starterdb.SID=rac
oracle.install.db.config.starterdb.characterSet=AL32UTF8
oracle.install.db.config.starterdb.memoryOption=true
oracle.install.db.config.starterdb.memoryLimit=2000
oracle.install.db.config.starterdb.installExampleSchemas=true
oracle.install.db.config.starterdb.enableSecuritySettings=false
oracle.install.db.config.starterdb.password.ALL=xxxx
oracle.install.db.config.starterdb.control=DB_CONTROL
oracle.install.db.config.starterdb.automatedBackup.enable=false
oracle.install.db.config.starterdb.storageType=FILE_SYSTEM_STORAGE
oracle.install.db.config.starterdb.fileSystemStorage.dataLocation=/oradata
oracle.install.db.config.asm.ASMSNMPPassword=xxxx
SECURITY_UPDATES_VIA_MYORACLESUPPORT=false
DECLINE_SECURITY_UPDATES=true
oracle.installer.autoupdates.option=SKIP_UPDATES
```

Sample script for preoracle.sh

The sample shell `preporacle.sh` script file is used for the first time to customize a Linux environment for Oracle.

The script is used in a second step to start the Oracle `runInstaller` program.

Because it updates the Linux system configuration files, such as, `sysctl.conf`, `limits.conf`, or ssh scripts, the file must be carefully tested before it is implemented in a production, as shown in the following example:

```
#!/bin/sh
#
# preporacle.sh
#
# Script shell used to prepare Oracle Single Instance environment on Linux
System z
# for Oracle 11G Release and launch a silent installation
# C. Noirault - A. Cheve
# 2012, September 17th
# Must be carefully tested before implementing in a production.
#
# First section must be configured with your variables.
# Second section performs:
#  Groups and Users Creation
#  Update the Oracle profile
#  Copies the silent install script
#  Test and create if needed /usr/local/bin
#  Create directories needed to install Oracle
#  Update kernel parameters in /etc/sysctl.conf
#  Update /etc/limits.conf
#  Update /etc/pam.d/sshd
#  Install rpms for Oracle
#  Prepare a vncserver to avoid being prompted when starting vncserver before
Oracle Install
# Third section launches Silent runInstaller
#
# The mounted NFS server should contain unzipped Oracle code packages:
#  Oracle code in /mnt/database
#  Custom Admin scripts in /mnt/script, including silent install and rpms
#
# Customize below with your own names, group and paths
#######################################################
# Oracle user definitions
#set -x
ORACLE_INSTALL_UID=1100
ORACLE_INSTALL_USER=oracle
ORACLE_INSTALL_GROUP=oinstall
# Groups definitions
ORACLE_INSTALL_GID=1100
ORACLE_DBA_GID=1101
ORACLE_OPER_GID=1102
# Directories definitions
ORACLE_USER_HOME=/home/$ORACLE_INSTALL_USER
```

```
ORACLE_ROOT=/u01/oracle
ORACLE_INVENTORY=/u01/oraInventory
ORACLE_SOURCE_NFS=l5ntcdom.mop.ibm.com:/drivers
ORACLE_INSTALL_MNT=/mnt
ORACLE_SI_SCRIPT=db-install-11.2.0.3-SLES11SP1.rsp
ORACLE_DATA=/$ORACLE_ROOT/oradata
ORAOUTPUT=/tmp/preporacle.out
#######################################################
echo "Starting Oracle Environment Customization" > $ORAOUTPUT
echo
"###############################################################################
##" >> $ORAOUTPUT
echo "Setting the environment with the variables below:" >> $ORAOUTPUT
echo "Oracle UserId = " $ORACLE_INSTALL_UID >> $ORAOUTPUT
echo "Oracle Name = " $ORACLE_INSTALL_USER >> $ORAOUTPUT
echo "Oracle Group Name = " $ORACLE_INSTALL_GROUP >> $ORAOUTPUT
echo "Oracle GroupId = " $ORACLE_INSTALL_GID >> $ORAOUTPUT
echo "DBA GroupId = " $ORACLE_DBA_GID >> $ORAOUTPUT
echo "OPERATOR GroupId= " $ORACLE_OPER_GID >> $ORAOUTPUT
echo "Oracle Linux Home Directory = " $ORACLE_USER_HOME >> $ORAOUTPUT
echo "Oracle Installation Base Directory = " $ORACLE_ROOT >> $ORAOUTPUT
echo "Oracle Inventory Referential Directory = " $ORACLE_INVENTORY >>
$ORAOUTPUT
echo "Oracle RAC Home Directory = " $ORACLE_RAC >> $ORAOUTPUT
#######################################################
#
# Unmount/mount nfs source directory
#
#######################################################
cd /
umount /mnt 1>/dev/null 2>&1
mount -t nfs $ORACLE_SOURCE_NFS /mnt >> $ORAOUTPUT
vmount=`ls -l /mnt | wc -l`
if [ $vmount -lt 2 ] ; then
  echo "Oracle Source Mounted Filesystem is empty !"  >> $ORAOUTPUT
  #return -1
else
#######################################################
#
# Groups creation
# oracle
vgroup=`cat /etc/group | grep oinstall | wc -l `
if [ $vgroup -ne 1 ] ; then
   echo "creation du groupe oinstall" >> $ORAOUTPUT
   groupadd -g $ORACLE_INSTALL_GID $ORACLE_INSTALL_GROUP
fi
echo "group oinstall OK" >> $ORAOUTPUT
# oracle dba
vgroup=`cat /etc/group | grep dba | wc -l `
if [ $vgroup -ne 1 ] ; then
        echo "creation du groupe oinstall" >> $ORAOUTPUT
        groupadd -g $ORACLE_DBA_GID dba
fi
echo "group dba OK"  >> $ORAOUTPUT
# oracle operator
```

```
vgroup=`cat /etc/group | grep oper | wc -l `
if [ $vgroup -ne 1 ] ; then
    echo "creation du groupe oper" >> $ORAOUTPUT
   groupadd -g $ORACLE_OPER_GID oper
fi
echo "group oper OK" >> $ORAOUTPUT
#
# Test the oracle user existence. Add oracle user if not there
USERORACLE=`cat /etc/passwd | grep $ORACLE_INSTALL_USER | wc -l`
echo " cat /etc/passwd | grep $ORACLE_INSTALL_USER | wc -l =>"$USERORACLE"<="
>> $ORAOUTPUT
if [ $USERORACLE -ne 1 ] ; then
   useradd -u $ORACLE_INSTALL_UID -g $ORACLE_INSTALL_GID -G oinstall,dba,oper
-s /bin/ksh -m $ORACLE_INSTALL_USER
  if [ !  -d $ORACLE_USER_HOME ]; then
    # Create the oracle user home directory and change directory permission
bits
    echo "Creating "$ORACLE_USER_HOME >> $ORAOUTPUT
    mkdir $ORACLE_USER_HOME
    chown -R $ORACLE_INSTALL_USER":"$ORACLE_INSTALL_GROUP $ORACLE_USER_HOME
  fi
fi
#
# Update oracle user .profile
cd $ORACLE_USER_HOME
echo "Saving "$ORACLE_USER_HOME"/.profile into
"$ORACLE_USER_HOME"/.profile.back" >> $ORAOUTPUT
cp .profile .profile.back
echo "# oracle " >>.profile
echo "export TMP=/tmp" >>.profile
echo "export TMPDIR="\$TMP >>.profile
echo "export ORACLE_HOSTNAME=$HOSTNAME" >>.profile
echo "export ORACLE_BASE="$ORACLE_ROOT >>.profile
echo "export DB_HOME=\$ORACLE_BASE/product/11.2.0/dbhome_1" >>.profile
echo "export ORACLE_HOME=\$DB_HOME" >>.profile
echo "export PATH=\$PATH:\$ORACLE_HOME/bin" >>.profile
echo "export LD_LIBRARY_PATH=\$ORACLE_HOME/lib:/lib:/usr/lib" >>.profile
echo "#export ORACLE_SID=RAC1" >>.profile
echo "#export ORACLE_TERM=xterm" >>.profile

# DISPLAY
echo "export DISPLAY=:1.0" >>.profile
# set hard limits for Oracle
echo "ulimit -n 65535" >>.profile
echo "ulimit -u 16384" >>.profile
echo "ulimit -s 32768" >>.profile
echo "# End oracle "  >>.profile
chown -R $ORACLE_INSTALL_USER":"$ORACLE_INSTALL_GROUP $ORACLE_USER_HOME
#
# Change Oracle user password
echo "oracle user is created. Change oracle password please"
#passwd oracle
#
# Copy script to Oracle home
cp $ORACLE_INSTALL_MNT/script/$ORACLE_SI_SCRIPT $ORACLE_USER_HOME
```

```
echo "cp "$ORACLE_INSTALL_MNT/script/$ORACLE_SI_SCRIPT $ORACLE_USER_HOME >>
$ORAOUTPUT
# Customize the script contents with new system hostname
echo "s/ORACLE_HOSTNAME=.*/ORACLE_HOSTNAME=`hostname --long`/" >
/tmp/silentscript1.tmp
cat /tmp/silentscript1.tmp >> $ORAOUTPUT
sed -f "/tmp/silentscript1.tmp" "$ORACLE_INSTALL_MNT/script/$ORACLE_SI_SCRIPT"
> /tmp/silentscript2.tmp
grep "ORACLE_HOSTNAME" /tmp/silentscript2.tmp >> $ORAOUTPUT
mv /tmp/silentscript2.tmp $ORACLE_USER_HOME/$ORACLE_SI_SCRIPT >> $ORAOUTPUT
rm /tmp/silentscript1.tmp >> $ORAOUTPUT
# Change script permission bits
chown $ORACLE_INSTALL_USER":"$ORACLE_INSTALL_GROUP
$ORACLE_USER_HOME/$ORACLE_SI_SCRIPT
echo "chown "$ORACLE_INSTALL_USER":"$ORACLE_INSTALL_GROUP
$ORACLE_USER_HOME/$ORACLE_SI_SCRIPT >> $ORAOUTPUT
#
# Test and create if needed /usr/local/bin
# Change /usr/local/bin directory permission bits to avoid being prompted by
Oracle install program
if [ !  -d /usr/local/bin ]; then
  echo "creation /usr/local" >> $ORAOUTPUT
  mkdir /usr/local
  mkdir /usr/local/bin
  chmod 777 /usr/local/bin
  chown bin /usr/local/bin
  chgrp bin /usr/local/bin
fi
#
#cp /usr/bin/ssh-keygen /usr/local/bin/ssh-keygen
#cp /usr/bin/ssh /usr/local/bin/ssh
#
# Create /u01/oracle directory to handle Oracle product
if [ ! -d $ORACLE_ROOT ] ; then
  mkdir -p $ORACLE_ROOT
  chown -R $ORACLE_INSTALL_USER":"$ORACLE_INSTALL_GROUP $ORACLE_ROOT
fi
# Create /u01/oraInventory directory to handle Oracle referentiel
if [ ! -d $ORACLE_INVENTORY ] ; then
  mkdir -p $ORACLE_INVENTORY
  chown -R $ORACLE_INSTALL_USER":"$ORACLE_INSTALL_GROUP $ORACLE_INVENTORY
fi
# Create /u01/oracle/oradata directory for DB installation
if [ ! -d $ORACLE_DATA ] ; then
   mkdir -p $ORACLE_DATA
   chown -R $ORACLE_INSTALL_USER":"$ORACLE_INSTALL_GROUP $ORACLE_DATA
fi
#
# Update kernel parameters in /etc/sysctl.conf
echo "# Oracle Kernel Specific parameters" >> /etc/sysctl.conf
echo "Saving /etc/sysctl.conf into /etc/sysctls.conf.back" >> $ORAOUTPUT
cp /etc/sysctl.conf /etc/sysctl.conf.back
# On SUSE Linux Enterprise Server systems only, you must enter the GID of the
oinstall group as the value for the parameter /proc/sys/vm/hugetlb_shm_ group.
```

```
Doing this grants members of oinstall a group permission to create shared
memory segments.
#
echo $ORACLE_INSTALL_GID > /proc/sys/vm/hugetlb_shm_group
#
echo 'BEGIN {' >/tmp/sysctl.txt
echo 'vkernelsem=0 ' >>/tmp/sysctl.txt
echo 'vkernelshmall=0 ' >>/tmp/sysctl.txt
echo 'vkernelsem=0 ' >>/tmp/sysctl.txt
echo 'vkernelshmall=0 ' >>/tmp/sysctl.txt
echo 'vkernelshmmax=0 ' >>/tmp/sysctl.txt
echo 'vkernelshmmni=0 ' >>/tmp/sysctl.txt
echo 'vfsfilemax=0 ' >>/tmp/sysctl.txt
echo 'vfsaiomaxnr=0 ' >>/tmp/sysctl.txt
echo 'vnetipv4iplocalportrange=0 ' >>/tmp/sysctl.txt
echo 'vnetcorermemdefault=0 ' >>/tmp/sysctl.txt
echo 'vnetcorermemmax=0 ' >>/tmp/sysctl.txt
echo 'vnetcorewmemdefault=0 ' >>/tmp/sysctl.txt
echo 'vnetcorewmemmax=0 ' >>/tmp/sysctl.txt
echo 'vkernelspinretry=0 ' >>/tmp/sysctl.txt
echo 'vmhugetlbshmgroup=0 ' >>/tmp/sysctl.txt
echo '} ' >>/tmp/sysctl.txt
#
echo '{' >> /tmp/sysctl.txt
echo 'vline=$0' >> /tmp/sysctl.txt
echo 'vOra_kernel=$1 " " $2' >> /tmp/sysctl.txt
echo 'if  ( $1 ~ /^kernel.sem/ ) {' >> /tmp/sysctl.txt
echo 'if ( $3 > 250 ) { vOra_kernel = vOra_kernel " " $3 }' >> /tmp/sysctl.txt
echo '  else { vOra_kernel = vOra_kernel " " 250 }' >> /tmp/sysctl.txt
echo '        if ( $4 > 32000 ) { vOra_kernel = vOra_kernel " " $4 }' >>
/tmp/sysctl.txt
echo '         else { vOra_kernel = vOra_kernel " " 32000 }' >> /tmp/sysctl.txt
echo '        if ( $5 > 100 ) { vOra_kernel = vOra_kernel " " $5 }' >>
/tmp/sysctl.txt
echo '         else { vOra_kernel = vOra_kernel " " 100 }' >> /tmp/sysctl.txt
echo '        if ( $6 > 128 ) { vOra_kernel = vOra_kernel " " $6 }' >>
/tmp/sysctl.txt
echo '         else { vOra_kernel = vOra_kernel " " 128 }' >> /tmp/sysctl.txt
echo '      vline=vOra_kernel' >> /tmp/sysctl.txt
echo '      vkernelsem=1 ' >> /tmp/sysctl.txt
echo ' }' >> /tmp/sysctl.txt
#
echo 'if  ( $1 ~ /^kernel.shmall/ ) {' >> /tmp/sysctl.txt
echo '  if ( $3 > 2097152 ) { vOra_kernel = vOra_kernel " " $3 }' >>
/tmp/sysctl.txt
echo '   else { vOra_kernel = vOra_kernel " " 2097152 }' >> /tmp/sysctl.txt
echo '  vline=vOra_kernel' >> /tmp/sysctl.txt
echo '  vkernelshmall=1 ' >> /tmp/sysctl.txt
echo ' }' >> /tmp/sysctl.txt
#
echo 'if  ( $1 ~ /^kernel.shmmax/ ) {' >> /tmp/sysctl.txt
echo '  if ( $3 > 4218210304 ) { vOra_kernel = vOra_kernel " " $3 }' >>
/tmp/sysctl.txt
echo '   else { vOra_kernel = vOra_kernel " " 4218210304 }' >> /tmp/sysctl.txt
echo '  vline=vOra_kernel' >> /tmp/sysctl.txt
```

```
echo '  vkernelshmmax=1 ' >> /tmp/sysctl.txt
echo ' }' >> /tmp/sysctl.txt
#
echo 'if  ( $1 ~ /^kernel.shmmni/ ) {' >> /tmp/sysctl.txt
echo '  if ( $3 > 4096 ) { vOra_kernel = vOra_kernel " " $3 }' >>
/tmp/sysctl.txt
echo '   else { vOra_kernel = vOra_kernel " " 4096 }' >> /tmp/sysctl.txt
echo '  vline=vOra_kernel' >> /tmp/sysctl.txt
echo '  vkernelshmmni=1 ' >>/tmp/sysctl.txt
echo ' }' >> /tmp/sysctl.txt
#
echo 'if  ( $1 ~ /^fs.file-max/ ) {' >> /tmp/sysctl.txt
echo '  if ( $3 > 6815744 ) { vOra_kernel = vOra_kernel " " $3 }' >>
/tmp/sysctl.txt
echo '   else { vOra_kernel = vOra_kernel " " 6815744 }' >> /tmp/sysctl.txt
echo '  vline=vOra_kernel' >> /tmp/sysctl.txt
echo '  vfsfilemax=1 ' >>/tmp/sysctl.txt
echo ' }' >> /tmp/sysctl.txt
#
echo 'if  ( $1 ~ /^fs.aio-max-nr/ ) {' >> /tmp/sysctl.txt
echo '  if ( $3 > 1048576 ) { vOra_kernel = vOra_kernel " " $3 }' >>
/tmp/sysctl.txt
echo '   else { vOra_kernel = vOra_kernel " " 1048576 }' >> /tmp/sysctl.txt
echo '  vline=vOra_kernel' >> /tmp/sysctl.txt
echo '  vfsaiomaxnr=1 ' >>/tmp/sysctl.txt
echo ' }' >> /tmp/sysctl.txt
#
echo 'if  ( $1 ~ /^net.ipv4.ip_local_port_range/ ) {' >> /tmp/sysctl.txt
echo '  if ( $3 > 9000 ) { vOra_kernel = vOra_kernel " " $3 }' >>
/tmp/sysctl.txt
echo '   else { vOra_kernel = vOra_kernel " " 9000 }' >> /tmp/sysctl.txt
echo '  if ( $4 > 65500 ) { vOra_kernel = vOra_kernel " " $4 }' >>
/tmp/sysctl.txt
echo '   else { vOra_kernel = vOra_kernel " " 65500 }' >> /tmp/sysctl.txt
echo '  vline=vOra_kernel' >> /tmp/sysctl.txt
echo '  vnetipv4iplocalportrange=1 ' >>/tmp/sysctl.txt
echo ' }' >> /tmp/sysctl.txt
#
echo 'if  ( $1 ~ /^net.core.rmem_default/ ) {' >> /tmp/sysctl.txt
echo '  if ( $3 > 262144 ) { vOra_kernel = vOra_kernel " " $3 }' >>
/tmp/sysctl.txt
echo '   else { vOra_kernel = vOra_kernel " " 262144 }' >> /tmp/sysctl.txt
echo '  vline=vOra_kernel' >> /tmp/sysctl.txt
echo '  vnetcorermemdefault=1 ' >>/tmp/sysctl.txt
echo ' }' >> /tmp/sysctl.txt
#
echo 'if  ( $1 ~ /^net.core.rmem_max/ ) {' >> /tmp/sysctl.txt
echo '  if ( $3 > 4194304 ) { vOra_kernel = vOra_kernel " " $3 }' >>
/tmp/sysctl.txt
echo '   else { vOra_kernel = vOra_kernel " " 4194304 }' >> /tmp/sysctl.txt
echo '  vline=vOra_kernel' >> /tmp/sysctl.txt
echo '  vnetcorermemmax=1 ' >>/tmp/sysctl.txt
echo ' }' >> /tmp/sysctl.txt
#
echo 'if  ( $1 ~ /^net.core.wmem_default/ ) {' >> /tmp/sysctl.txt
```

```
echo '  if ( $3 > 262144 ) { vOra_kernel = vOra_kernel " " $3 }' >>
/tmp/sysctl.txt
echo '   else { vOra_kernel = vOra_kernel " " 262144 }' >> /tmp/sysctl.txt
echo '  vline=vOra_kernel' >> /tmp/sysctl.txt
echo '  vnetcorewmemdefault=1 ' >>/tmp/sysctl.txt
echo ' }' >> /tmp/sysctl.txt
#
echo 'if  ( $1 ~ /^net.core.wmem_max/ ) {' >> /tmp/sysctl.txt
echo '  if ( $3 > 1048576 ) { vOra_kernel = vOra_kernel " " $3 }' >>
/tmp/sysctl.txt
echo '   else { vOra_kernel = vOra_kernel " " 1048576 }' >> /tmp/sysctl.txt
echo '  vline=vOra_kernel' >> /tmp/sysctl.txt
echo '  vnetcorewmemmax=1 ' >>/tmp/sysctl.txt
echo ' }' >> /tmp/sysctl.txt
#
echo 'if  ( $1 ~ /^kernel.spin_retry/ ) {' >> /tmp/sysctl.txt
echo '  if ( $3 > 2000 ) { vOra_kernel = vOra_kernel " " $3 }' >>
/tmp/sysctl.txt
echo '   else { vOra_kernel = vOra_kernel " " 2000 }' >> /tmp/sysctl.txt
echo '  vline=vOra_kernel' >> /tmp/sysctl.txt
echo '  vkernelspinretry=1 ' >>/tmp/sysctl.txt
echo ' }' >> /tmp/sysctl.txt
#
echo 'if  ( $1 ~ /^vm.hugetlb_shm_group/ ) {' >> /tmp/sysctl.txt
echo '  vline = "vm.hugetlb_shm_group =  "' $ORACLE_INSTALL_GID  >>
/tmp/sysctl.txt
echo '  vmhugetlbshmgroup=1 ' >>/tmp/sysctl.txt
echo ' }' >> /tmp/sysctl.txt

echo 'print vline' >> /tmp/sysctl.txt
echo '}' >> /tmp/sysctl.txt
echo 'END { '  >> /tmp/sysctl.txt
echo ' print "# Oracle parameters"; ' >>  /tmp/sysctl.txt
echo ' if ( vkernelsem=0 ) { print "kernel.sem = 250 32000 100 128" } ' >>
/tmp/sysctl.txt
echo ' if ( vkernelshmall=0 ) { print "kernel.shmall = 2097152" } ' >>
/tmp/sysctl.txt
echo ' if ( vkernelshmmax=0 ) { print "kernel.shmmax = 4218210304" } ' >>
/tmp/sysctl.txt
echo ' if ( vkernelshmmni=0 ) { print "kernel.shmmni = 4096" } ' >>
/tmp/sysctl.txt
echo ' if ( vfsfilemax=0 ) { print "fs.file-max = 6815744" } ' >>
/tmp/sysctl.txt
echo ' if ( vfsaiomaxnr=0 ) { print "fs.aio-max-nr = 1048576" } ' >>
/tmp/sysctl.txt
echo ' if ( vnetipv4iplocalportrange=0 ) { print "net.ipv4.ip_local_port_range
= 9000 65500" } ' >>  /tmp/sysctl.txt
echo ' if ( vnetcorermemdefault=0 ) { print "net.core.rmem_default = 262144" }
' >>  /tmp/sysctl.txt
echo ' if ( vnetcorermemmax=0 ) { print "net.core.rmem_max = 4194304" } ' >>
/tmp/sysctl.txt
echo ' if ( vnetcorewmemdefault=0 ) { print "net.core.wmem_default = 262144" }
' >>  /tmp/sysctl.txt
echo ' if ( vnetcorewmemmax=0 ) { print "net.core.wmem_max = 1048576" } ' >>
/tmp/sysctl.txt
```

```
echo ' if ( vkernelspinretry=0 ) { print "kernel.spin_retry = 2000" } ' >>
/tmp/sysctl.txt
echo ' if ( vmhugetlbshmgroup=0 ) { print "vm.hugetlb_shm_group = "'
$ORACLE_INSTALL_GID } >> /tmp/sysctl.txt
echo 'print "# End of Oracle parameters";  ' >>  /tmp/sysctl.txt
echo '} ' >> /tmp/sysctl.txt
#
awk -f /tmp/sysctl.txt /etc/sysctl.conf > /tmp/sysctl.tmp
#
cp /tmp/sysctl.tmp /etc/sysctl.conf
# The following command changes the current values of the kernel parameters
/sbin/sysctl -p  1>/dev/null 2>&1
# Enter the command /sbin/sysctl -a to confirm that the values are set
correctly
echo "........................................................."  >> $ORAOUTPUT
/sbin/sysctl -a  1>/dev/null 2>&1
echo "........................................................."  >> $ORAOUTPUT
# On SUSE Linux Enterprise Server systems only, enter the following command to
cause the system to read the /etc/sysctl.conf file when it restarts:
/sbin/chkconfig boot.sysctl on  1>/dev/null 2>&1
#
#/etc/limits.conf
#limite=`cat /etc/security/limits.conf | grep oracle | wc -l `
limite=`cat /etc/security/limits.conf | grep $ORACLE_INSTALL_USER | wc -l `
echo "cat /etc/security/limits.conf | grep "$ORACLE_INSTALL_USER" | wc -l
=>"$limite"<=" >> $ORAOUTPUT
echo "........................................................."  >> $ORAOUTPUT
if [ $limite -eq 0 ] ; then
  echo "# oracle "  >>/etc/security/limits.conf
  echo $ORACLE_INSTALL_USER" soft nofile 1024"  >>/etc/security/limits.conf
  echo $ORACLE_INSTALL_USER" hard nofile 65536"  >>/etc/security/limits.conf
  echo $ORACLE_INSTALL_USER" soft nproc 2047"  >>/etc/security/limits.conf
  echo $ORACLE_INSTALL_USER" hard nproc 16384"  >>/etc/security/limits.conf
  echo $ORACLE_INSTALL_USER" soft stack 10240"  >>/etc/security/limits.conf
  echo $ORACLE_INSTALL_USER" hard stack 32768"  >>/etc/security/limits.conf
  echo "# Use memlock for Huge Pages support"  >>/etc/security/limits.conf
  echo "#*softmemlock 3145728"  >>/etc/security/limits.conf
  echo "#*hardmemlock 3145728"  >>/etc/security/limits.conf
  echo "# End oracle "  >> /etc/security/limits.conf
fi
#
#/etc/pam.d/sshd
limite=`cat /etc/pam.d/sshd | grep pam_limits.so | wc -l`
echo "cat /etc/pam.d/sshd | grep pam_limits.so | wc -l =>"$limite"<=" >>
$ORAOUTPUT
echo "........................................................."  >> $ORAOUTPUT
if [ $limite -eq 0 ] ; then
  echo "Saving /etc/pam.d/sshd into /etc/pam.d/sshd.back" >> $ORAOUTPUT
  cp /etc/pam.d/sshd /etc/pam.d/sshd.back
  echo "# Oracle "  >> /etc/security/limits.conf
  echo "session required pam_limits.so"  >>/etc/pam.d/sshd
  echo "# End oracle "  >> /etc/security/limits.conf
fi
# rpms for Oracle
orarpm=`rpm -qa | grep ora-val | wc -l`
```

```
echo "rpm -qa | grep ora-val | wc -l =>"$limite"<=" >> $ORAOUTPUT
echo "...................................................."  >> $ORAOUTPUT
if [ $orarpm -eq 0 ] ; then
  echo "Installing Oracle rpm" >> $ORAOUTPUT
  echo "rpm -ivh "$ORACLE_INSTALL"/ora-val-rpm-S11-DB-11.2.0.2-1.s390x.rpm" >>
$ORAOUTPUT
  rpm -ivh $ORACLE_INSTALL/ora-val-rpm-S11-DB-11.2.0.2-1.s390x.rpm
fi
orarpm=`rpm -qa | grep cvuqdisk | wc -l`
echo "rpm -qa | grep cvuqdisk | wc -l =>"$limite"<=" >> $ORAOUTPUT
echo "...................................................."  >> $ORAOUTPUT
if [ $orarpm -eq 0 ] ; then
  echo "Installing Oracle cvuqdisk rpm" >> $ORAOUTPUT
  echo "rpm -iv "$ORACLE_INSTALL"/cvuqdisk-1.0.9-1.rpm" >> $ORAOUTPUT
  rpm -iv $ORACLE_INSTALL/cvuqdisk-1.0.9-1.rpm
fi
#
##################################################################
# Prepare a vncserver to avoid being prompted when starting vncserver before
Oracle Install
# Needed to run host+
##################################################################
if [ ! -d /root/.vnc ] ; then
   mkdir -p /root/.vnc  >> $ORAOUTPUT
   cd /root/.vnc
   # build a passwd file with password="password"
   echo "begin 600 passwd" > passwd2
   echo "(V]@\_7)Z%%@\`" >> passwd2
   echo "\`"  >> passwd2
   echo "end" >> passwd2
   # generate passwd withh uudecode, passwd2 being the result of
   #  uuencode passwd passwd > passwd2
   uudecode passwd2
   # build a xstartup file
   echo "#!/bin/sh" > xstartup
   echo ""           >> xstartup
   echo "xrdb \$HOME/.Xresources" >> xstartup
   echo "xsetroot -solid grey" >> xstartup
   echo "xterm -geometry 80x24+10+10 -ls -title \"\$VNCDESKTOP Desktop\" &" >>
xstartup
   echo "twm &" >> xstartup
fi
chown -R  root:root /root/.vnc
vncserver
xhost +
cvnc=`ps -ef | grep vnc`
#
##################################################################
# run the Installer
##################################################################
# Launch runInstaller
# runInstaller  can run with -showProgress parameter to show installation
progress, useless in automated case
echo "Executing runInstaller."
export ORACLE_INSTALL_MNT
```

```
export ORACLE_INSTALL_USER
export ORACLE_USER_HOME
export ORACLE_SI_SCRIPT
echo "su -c \"cd /home/$ORACLE_INSTALL_USER;. ./.profile;cd
$ORACLE_INSTALL_MNT/database; ./runInstaller -silent -ignorePrereq  -force
-responseFile $ORACLE_USER_HOME/$ORACLE_SI_SCRIPT\" $ORACLE_INSTALL_USER" >>
$ORAOUTPUT
su -c "cd /home/$ORACLE_INSTALL_USER;. ./.profile;cd
$ORACLE_INSTALL_MNT/database; ./runInstaller -silent -ignorePrereq  -force
-responseFile $ORACLE_USER_HOME/$ORACLE_SI_SCRIPT" $ORACLE_INSTALL_USER
echo "End runInstaller."
#
#################################################################
# VERIFICATIONS
#################################################################
echo
"##############################################################################
##" >> $ORAOUTPUT
echo "Environment variables set:" >> $ORAOUTPUT
echo "Oracle UserId = " $ORACLE_INSTALL_UID >> $ORAOUTPUT
echo "Oracle Name = " $ORACLE_INSTALL_USER >> $ORAOUTPUT
echo "Oracle Group Name = " $ORACLE_INSTALL_GROUP >> $ORAOUTPUT
echo "Oracle GroupId = " $ORACLE_INSTALL_GID >> $ORAOUTPUT
echo "DBA GroupId = " $ORACLE_DBA_GID >> $ORAOUTPUT
echo "OPERATOR GroupId= " $ORACLE_OPER_GID >> $ORAOUTPUT
echo "Oracle Linux Home Directory = " $ORACLE_USER_HOME >> $ORAOUTPUT
echo "Oracle Installation Base Directory = " $ORACLE_ROOT >> $ORAOUTPUT
echo "Oracle Inventory Referential Directory = " $ORACLE_INVENTORY >>
$ORAOUTPUT
#################################################################
echo
"##############################################################################
##" >> $ORAOUTPUT
echo "Verif directories creation" >> $ORAOUTPUT
ls -al /u01 >> $ORAOUTPUT
echo
"##############################################################################
##" >> $ORAOUTPUT
echo "Verif user and groups creation" >> $ORAOUTPUT
cat /etc/group | grep oinstall  >> $ORAOUTPUT
cat /etc/group | grep dba  >> $ORAOUTPUT
cat /etc/group | grep oper  >> $ORAOUTPUT
cat /etc/passwd | grep $ORACLE_INSTALL_USER  >> $ORAOUTPUT
echo
"##############################################################################
##" >> $ORAOUTPUT
echo "Verif oracle home creation" >> $ORAOUTPUT
ls -al /home/$ORACLE_INSTALL_USER >> $ORAOUTPUT
cat /home/$ORACLE_INSTALL_USER/.profile >> $ORAOUTPUT
echo
"##############################################################################
##" >> $ORAOUTPUT
echo "Verif /user/local/bin chmod" >> $ORAOUTPUT
ls -l /usr/local >> $ORAOUTPUT
```

```
echo
"###############################################################################
##" >> $ORAOUTPUT
echo "Verif /etc/sysctl.conf" >> $ORAOUTPUT
#cat /etc/sysctl.conf >> $ORAOUTPUT
echo
"###############################################################################
##" >> $ORAOUTPUT
echo "Verif /etc/security/limits.conf" >> $ORAOUTPUT
cat /etc/security/limits.conf | grep $ORACLE_INSTALL_USER >> $ORAOUTPUT
echo
"###############################################################################
##" >> $ORAOUTPUT
echo "Verif /etc/pam.d/sshd" >> $ORAOUTPUT
cat /etc/pam.d/sshd | grep  pam_limits.so >> $ORAOUTPUT
echo
"###############################################################################
##" >> $ORAOUTPUT
echo "Oracle Environment Customization End" >> $ORAOUTPUT
fi
#return 0
# End preporacle.sh
```

Sample Tivoli Provisioning Manager workflow

A sample Tivoli Provisioning Manager workflow is shown in the following example:

```
# -----------------------------------------------------------------
# Licensed Materials - Property of IBM
# 5724-F75
# (C) Copyright IBM Corp.  2003 - 2009
# All Rights Reserved
# US Government Users Restricted Rights -Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
# -----------------------------------------------------------------
@param DeviceID - DCM object ID of the target server
@doc Dependent Workflows:

workflow Oracle11203_Install(in DeviceID) LocaleInsensitive
  var Source_DeviceID =  5678
  var Source_ToolsDir = "/opt/ibm/oracle"
  var Dest_DeviceID = DeviceID
  var Dest_ToolsDir = "/tmp"
  var Script_Name = "./preporacle.sh"
  #Search key for client credentials to be used, if it is empty then default
Credential key will be used
  var ClientCredentialsKey = "default"
  var HostCredentialsKey = "default"
  var TimeoutInSeconds = 300
  #Copies the script file to the server
  log info Jython["ACH Start copying Script"]
  Device.CopyFile(Source_DeviceID, Source_ToolsDir, Script_Name, Dest_DeviceID,
Dest_ToolsDir, Script_Name, ClientCredentialsKey, HostCredentialsKey,
TimeoutInSeconds )
```

```
  log info Jython["ACH End copying  Script"]
  log info Jython["ACH Start executing script"]
  #Device.Execute Command
   #@param DeviceId DCM id of the device
   #@param ExecuteCommand The command to be executed including arguments
   #@param WorkingDirectory The path of the working directory
   #@param CredentialsKey Key for identifying credentials for accessing the
service
   #@param TimeoutInSeconds Number of seconds elapsed to stop waiting for
operation completion
   #@param TreatTimeoutAs Action when the command timeouts. Possible values:
ignore, warning, error.
   #@param ReturnCode Contains the return exit code in case of success, null in
case of failure
   #@param ReturnErrorString Content of error stream (stderr) in case of
success, null in case of failure
   #@param ReturnResult Return result (content of stdout) in case of success,
null in case of failure
Device.ExecuteCommand(DeviceID, Script_Name, Dest_ToolsDir,
ClientCredentialsKey, TimeoutInSeconds, "error", <null>, <null>, <null>)
log info Jython["ACH end executing script"]
```

References books

The following Redbooks publications are available:

- *z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 11 SP1*, SG24-7931-00
- *Directory Maintenance Facility Tailoring and Administration Guide; Version 6 Release 1*, SC24-6190-00
- *Deploying a Cloud on IBM System z*, REDP-4711
- *Provisioning Linux on IBM System z with Tivoli Service Automation Manager*, REDP-4663
- *Installing Oracle 11gR2 RAC on Linux on System z*, REDP-4788
- *Experiences with a Silent Install of Oracle Database 11gR2 RAC on Linux on System z (11.2.0.3)*, REDP-9131
- *Tivoli Service Automation Manager Version 7.2.2 - Installation and Administration Guide*, SC34-2657-00

For more information about IBM publications, see the Redbooks website at:

http://www.redbooks.ibm.com

For more information about Tivoli, see the “Configuring the z/VM environment for Tivoli Service Automation Manager” topic in the Tivoli Information Center at this website:

http://pic.dhe.ibm.com/infocenter/tivihelp/v10r1/index.jsp

Related publications

The publications that are listed in this section are considered particularly suitable for a more detailed discussion of the topics that are covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide more information about the topic in this document. Note that some publications that are referenced in this list might be available in softcopy only:

- *Experiences with Oracle Solutions on Linux for IBM System z, SG24-7634*
- *Silent Installation Experiences with Oracle Database 11gR2 Real Application Clusters on Linux on System z,* REDP-9131
- *Installing Oracle 11gR2 RAC on Linux on System z,* REDP-4788
- *Sharing and maintaining Linux under z/VM,* REDP-4322
- *Optimizing Your Oracle Investment with IBM Storage Solutions,* REDP-4421
- *An Introduction to z/VM Single System Image (SSI) and Live Guest Relocation (LGR),* SG24-8006
- *Using z/VM v 6.2 Single System Image (SSI) and Live Guest Relocation (LGR),* SG24-8039
- *Using IBM Virtualization to Manage Cost and Efficiency*, REDP-4527-00
- *Linux on IBM System z: Performance Measurement and Tuning,* SG24-6926

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft, and additional materials at the following website:

`http://www.ibm.com/redbooks`

Other publications

The following publications are also relevant as further information sources:

- *z/VM and Linux on IBM System z: The Virtualization Cookbook for z/VM 6.2 RHEL 6.2 and SLES 11 SP2,* which is available at the following websites:
  - `http://www.vm.ibm.com/devpages/mikemac/CKB-VM62.PDF`
  - `http://www.vm.ibm.com/devpages/mikemac/CKB-VM62.tgz`
- *z/VM and Linux on IBM System z: The Cloud Computing Cookbook for z/VM 6.2, RHEL 6.2 and SLES 11 SP2,* which is available at the following websites:
  - `http://www.vm.ibm.com/devpages/mikemac/CKBOVM62.PDF`
  - `http://www.vm.ibm.com/devpages/mikemac/CKB-VM62.tgz`
- *High-Availability of System Resources: Architectures for Linux on IBM System z Servers*:

  `http://public.dhe.ibm.com/common/ssi/ecm/en/zsw03236usen/ZSW03236USEN.PDF`

My Oracle Support notes

The following My Oracle Support notes are related to Oracle on Linux on System z:

- Note 1306465.1 Getting Started - 11gR2 Grid Infrastructure, SI (Single Instance), ASM and DB (IBM: Linux on System z)
- Note 1470834.1 Requirements for Installing Oracle 11gR2 on RHEL 6 on IBM: Linux on System z (s390x)
- Note 1290644.1 Requirements for Installing Oracle 11gR2 on SLES11 on IBM: Linux on System z (s390x) Also review note: OHASD fails to start on SuSE 11 SP2 on IBM: Linux on System z [ID 1476511.1]
- Note 1308859.1 Requirements for Installing Oracle 11gR2 on SLES 10 on IBM: Linux on System z (s390x)
- Note 1306889.1 Requirements for Installing Oracle 11gR2 on RHEL 5 on IBM: Linux on System z (s390x)
- Note 1086769.1 Ensure you have prerequisite rpms to install Oracle Database and AS10g(midtier) on IBM: Linux on System z (s390x)
- Note 1377392.1 How to Manually Configure Disk Storage devices for use with Oracle ASM 11.2 on IBM: Linux on System z Red Hat 6
- Note 1350008.1 How to Manually Configure Disk Storage devices for use with Oracle ASM 11.2 on IBM: Linux on System z SLES
- Note 1351746.1 How to Manually Configure Disk Storage devices for use with Oracle ASM 11.2 on IBM: Linux on System z Red Hat 5
- Note 1400185.1 How to Upgrade Oracle Restart i.e. Single Node Grid Infrastructure/ASM from 11.2.0.2 to 11.2.0.3
- Note 1276058.1 Oracle GoldenGate Best Practices: Instantiation from an Oracle Source Database
- Note 1413787.1 How to completely remove 11.2 Grid Infrastructure, CRS and/or Oracle Restart - IBM: Linux on System z
- Note 259301.1 CRS and 10g Real Application Clusters
- Note 268937.1 Repairing or Restoring an Inconsistent OCR in RAC
- Note 239998.1 10g RAC How to clean up after a failed CRS Install
- Note 220970.1 RAC Frequently Asked Questions Topic
- Note 420382.1 Requirements for Installing Oracle 10gR2 RDBMS on RHEL 4 on zLinux (s390x).
- Note 431443.1 Requirements for Installing Oracle 10gR2 RDBMS on SLES 9 zLinux (s390x)
- Note 1082253 Requirements for Installing Oracle 10gR2 RDBMS on SLES 10 zLinux (s390x)
- Note 741646.1 Requirements for Installing Oracle 10gR2 RDBMS on RHEL 5 on zLinux (s390x).
- Note 415182.1 DB Install Requirements Quick Reference - zSeries based Linux.
- Note 741146.1 Installing Standalone Agent 10.2 on Linux on z
- Note 431443.1 Requirements for Installing Oracle 10gR2 RDBMS on SLES 9 zLinux (s390x)

- Note 1082253 Requirements for Installing Oracle 10gR2 RDBMS on SLES 10 zLinux (s390x)
- Note 741646.1 Requirements for Installing Oracle 10gR2 RDBMS on RHEL 5 on zLinux (s390x).
- Note 415182.1 DB Install Requirements Quick Reference - zSeries based Linux.

Online resources

The following websites are also relevant as further information sources:

- Oracle Technology Network:

  http://otn.oracle.com
- Oracle Support web page (My Oracle Support):

  https://support.oracle.com
- Special Interest Group of Oracle users on the mainframe:

  http://www.zseriesoraclesig.org
- Linux on System z:

  http://www.ibm.com/developerworks/linux/linux390/
- z/VM Performance and Tuning Tips, Capacity planning:

  http://www.vm.ibm.com/perf/tips
- IBM Tivoli Service Automation Manager, Version 7.2.2, Setting up z/VM for Linux provisioning:

  http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/index.jsp?topic=%2Fcom.ibm.tsam.doc_7.2%2Ft_config_zvm_setup.html
- Hints and Tips for tuning Linux on System z:

  http://www.ibm.com/developerworks/linux/linux390/perf/index.html

Help from IBM

IBM Support and downloads:

http://www.ibm.com/support

IBM Global Services:

http://www.ibm.com/services

IBM

Redbooks

Experiences with Oracle 11gR2 on Linux on System z

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

Experiences with Oracle 11gR2 on Linux on System z

Installing Oracle 11gR2 on Linux on System z

Managing an Oracle environment

Provisioning an Oracle environment

Linux on System z offers many advantages to customers who rely on the IBM mainframe systems to run their businesses. Linux on System z makes use of the qualities of service in the System z hardware and in z/VM, making it a robust industrial strength Linux. This provides an excellent platform for hosting Oracle solutions that run in your enterprise.

This IBM Redbooks publication is divided into several sections to share the following experiences that are gained while Oracle Database 11gR2 is installed and tested:

- Setting up Red Hat Enterprise Linux 6 for Oracle
- Managing an Oracle on Linux on System z environment
- Provisioning Linux guests using several tools

It also includes many general hints and tips for running Oracle products on IBM System z with Linux and z/VM.

Interested readers include database consultants, installers, administrators, and system programmers. This book is not meant to replace Oracle documentation but to supplement it with our experiences while Oracle products are installed and used.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-8104-00 ISBN 0738438715