Migration Program Overview Best Practice

www.novell.com

RHEL to SLES Migration Overview

Published: Feb, 2009
Version # 1.3

Novell

Novell, Inc.
404 Wyman Street
Waltham
Massachusetts 02451
USA

Novell UK Ltd
1 Arlington Square
Bracknell
Berkshire
RG12 1WA
UK

Prepared By Daniel Westervelt, Rich Paredes, and Anthony Tortola

RHEL to SLES Migration Overview—Migration Program Overview Best Practice

Feb, 2009

RHEL to SLES Migration Overview

Contents

Tables

Figures

1 Executive Summary

Four components of a RHEL to SLES migration:

1. Distribution and Software Support
   - *Both RHEL and SLES must be equally supportable during the migration.*
2. Deployment Infrastructure
   - *Infrastructure must be in place to deploy both RHEL and SLES during migration.*
3. Patching and Maintenance
   - *Infrastructure must be in place to be able to patch and perform maintenance on both RHEL and SLES during migration.*
4. Application Portfolio
   - *The necessary development tools and application resources must be made available to migrate applications from RHEL to SLES without any loss of functionality or performance.*

2 Distribution and Software Support

2.1 Red Hat Enterprise Linux Patch Updates

Patch Availability

All RHEL patch updates are made available via a RHEL update channel through the Novell Customer Center.

Patch Release Cycle

Novell's RHEL patches will follow the current Red Hat release with an expected turnaround of 1 day from public posting. Patches associated with Severity 1 tickets will be escalated as required.

Patch Equivalence

The rpm update patches will maintain a payload that is built from the same sources as the actual Red Hat release. Additionally a Novell signature will be added to the package.

2.2 Novell RHEL Support Escalation

Support Commitments

- Patches will be made available for:
  - RHEL 3.9 and subsequent releases
  - RHEL 4.7 and subsequent releases
  - RHEL 5.2 and subsequent releases
- By default, patches are made available through Novell's SMT utility, although, Novell Zenworks can also be used if a more complete lifecycle management tool is required.*
- Each Red Hat release is supported in a manner similar to SLES with full level1, level2, and level3 support commitments.
- A portion of Novell's support staff are Red Hat certified. Novell is committed to continue to train our support team commensurate with the programs further adoption.
- Red Hat release media for new builds is not provided and/or supported as part of this program.

* Zenworks Linux Management requires an additional license purchase. It is not part of the standard support offering.

2.2.1 Novell RHEL Bug Fix Process

In the case that a bug is identified in RHEL that is not fixed in a currently available Red Hat patch, Novell will follow the process defined below:

1. Novell support engineers to begin working on a patch.
2. Novell will open a ticket in Red Hat's Bugzilla.
3. Novell will supply a patch to the client.
4. Novell will update Red Hat Bugzilla with the new bugfix.
5. Wait for response from Red Hat.
   a. Red Hat accepts patch.
   b. Red Hat produces their own bugfix.
6. Client has option of keeping Novell patch or updating with Red Hat version if it differs.

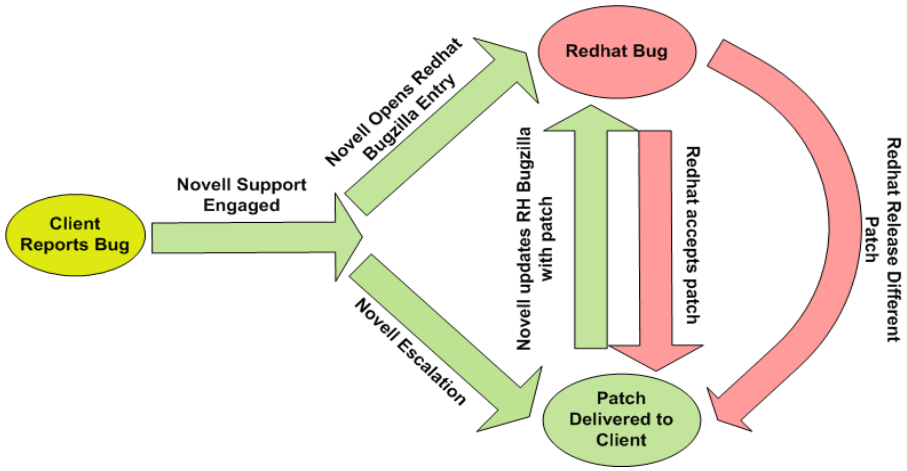

Figure 1: Bugfix Flowchart

2.3 Non-RHEL Red Hat Application Support

While many of the same tools and applications are available in both the RHEL and SLES distributions there are some notable differences. There are two possible approaches to supporting RHEL specific applications on SLES.

1. Migrate to a Novell alternative. For instance, Novell's eDirectory can easily import Red Hat Directory Server schema and data.
2. In some cases Novell can provide and support RHEL products. For instance, Novell can support Red Hat Cluster Services (RHCS), Global File System (GFS), and JBoss. In fact, Novell currently supports Red Hat Cluster Services and the Global File System on SLES10**.

   ** Currently supported for customers who are already running RHCS/GFS on RHEL or by special arrangement.

2.4 SLES Training

While RHEL and SLES are fundamentally based on the same open source components, Novell believes that there are enough technical differentiators in SLES that training is a core component to a successful migration. Depending on the level of experience within your internal engineering and support teams, Novell can recommend a customized training program to fit your needs.

Novell also provides on-line programs to assist in determining the specific training needs of your organization.

3 Deployment Infrastructure

3.1 Linux Build Environment

A Linux build environment consists of all technologies and media required to provision a server. Generally speaking there are two main methods of unattended provisioning; The first is via imaging, while the second is done via a template driven installation. Both are supported options in the SLES migration tool-set, each having its own advantages and disadvantages.

Template Driven Installations

Template driven system installations are generally based on Red Hat's Anaconda/Kickstart or Novell's AutoYaST. Each technology provisions a server's operating system based on a pre-defined template and a remotely accessible install media source. Novell's SLE migration supports both environments via a singular PXE configuration. Novell also provides tools to automatically analyze your current Kickstart templates and produce equivalent AutoYaST templates.

Image Driven Installations

Image driven system installations use a pre-installed system image that is copied to a new system's drive. Local configurations are then usually applied either manually or by some scripting component. While image driven installs are usually quick they typically suffer from the overhead of managing multiple images in a central repository, coupled with the complexity of post install configurations.

Novell supports image provisioning via Zenworks Linux Management (ZLM).

3.2 Novell SLES Migration Build QuickStart

In order to facilitate a rapid deployment environment for a SLES migration, Novell has developed a process to quickly establish a SLES remote build environment via our Quickstart utility. This utility is designed to automate the creation of a SLES remote build environment. Once established, this environment can be pointed to the current RHEL build environment in order to provide consistent mechanisms for any required server deployments.

3.3 Build Infrastructure

The more commonly implemented build environments are:

1. Data center DHCP

   The data center DHCP environment is where the majority of networks in the data center have access to a DHCP service. An example is where IPhelper is used to gain access to DHCP from multiple networks. This environment allows for servers to be provisioned in place.

2. Staging Lab

   In data centers where DHCP is not allowed across the network, a staging lab can be used to provision servers. This is generally an isolated network that can be safely served by the build DHCP. Servers are pre-provisioned on this network before being moved to their final location in a production data center.

A SLES migration build environment is designed to coexist with your current RHEL build environment. The combined environment generally looks like the one below:

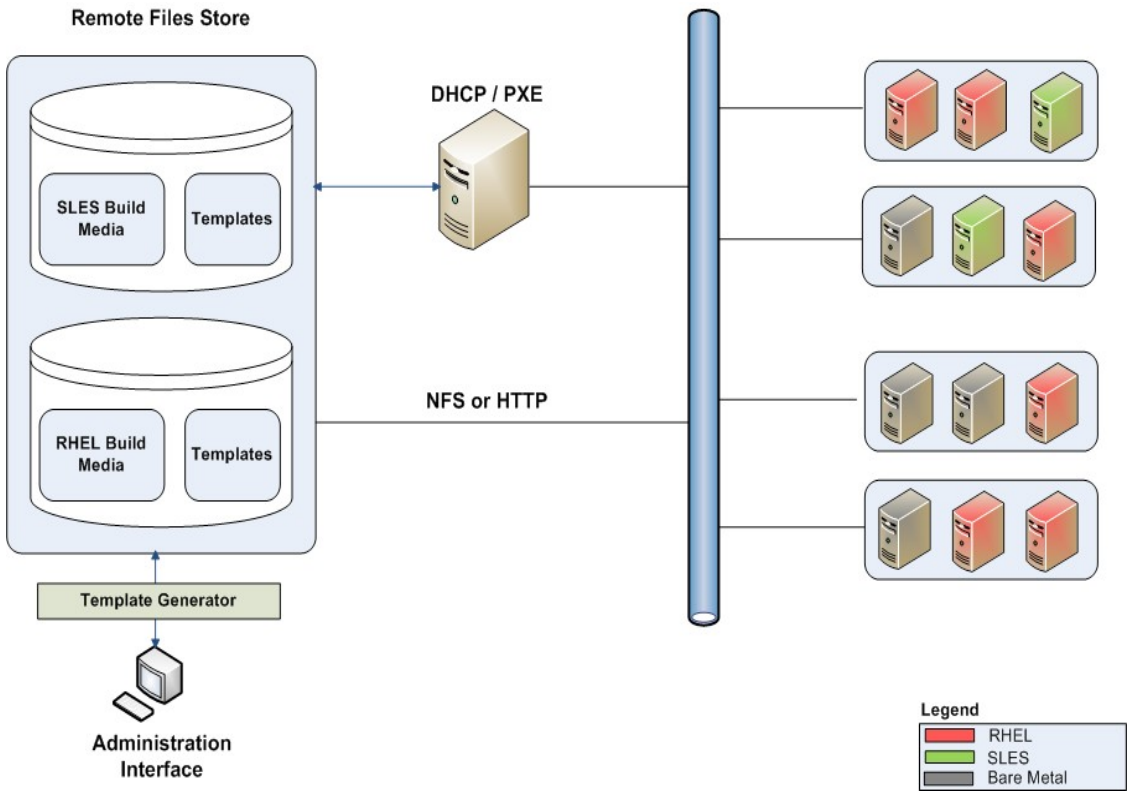

Figure 2: Mixed Build Environment

The RHEL and SLES build media are stored in different disk volumes or directories and are made available for access via HTTP or NFS. An administration interface is available for build template generation.

4 Patching and Maintenance

4.1 Patch Management

As part of a SLES migration, Novell offers patches for both RHEL and SLES via Novell's Customer Center. It is recommended that the new patch management system be deployed in parallel to your current patch system to ensure parity. As a comfort level is reached, the legacy patch system can be decommissioned.

4.2 System Inventory

A complete central inventory of Linux devices is essential to proper patching and maintenance during and after the migration process. This inventory, combined with an application inventory, is the map necessary to progress through the migration successfully. Novell can assist with this inventory collection and management in a number of ways, including:

Collecting System Inventory

1. Pulling data from pre-existing inventory system(s).
2. Leveraging Novell's own inventory tools for self discovery.

Managing System Inventory

1. Inventory management through ZENworks Linux Management or ZENworks Asset Management.
2. Integrating with existing inventory system(s).
3. Through project management oversight.

4.3 Patch Management Options

4.3.1 Subscription Management Tool (SMT)

Utilizes current existing patch protocols such as YUM, YaST or Red Hat's up2date for basic patching functionality. This is important as it does not require any additional software to be installed on your current Red Hat servers for patching. Some highlights of this solution include:

- Secure RHEL and SLES patch download from NCC.
- Pull patching via YUM, YaST and up2date.
- Can be scheduled via local cron.
- Freely available.

The figure below depicts what a base Subscription Management Tool (SMT) environment might look like.

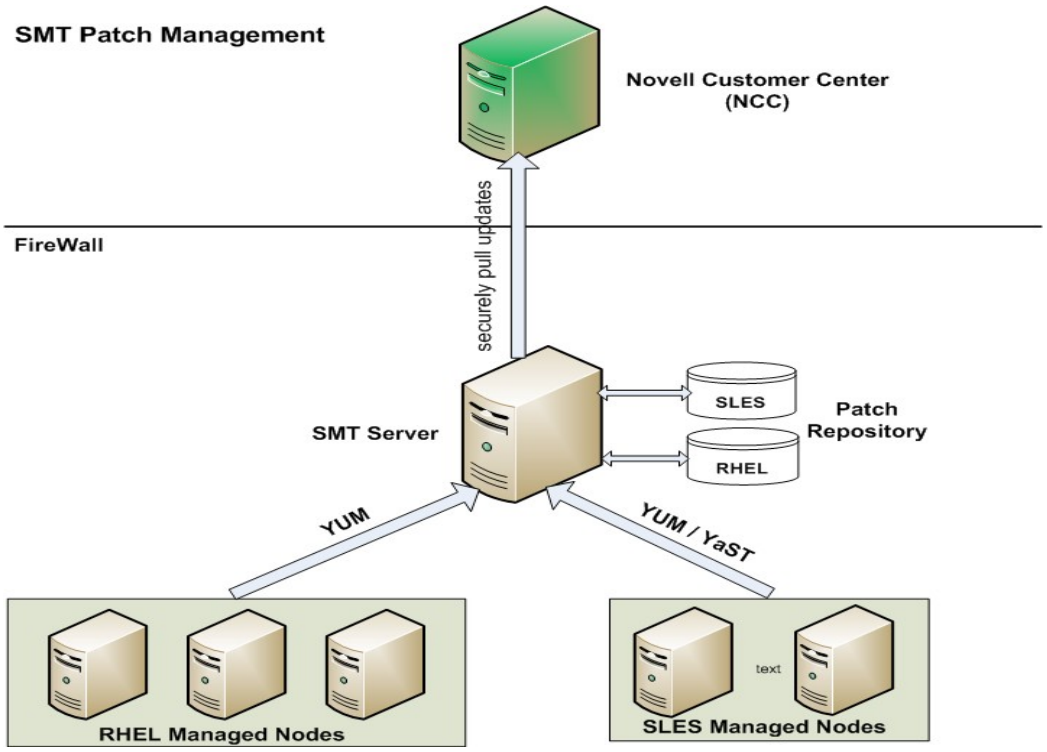

Figure 3: SMT Patch Management

4.3.2 ZENworks Linux Management (ZLM)

ZENworks Linux Management allows for the complete Linux life cycle management. This includes the ability to centrally manage your complete Linux hardware portfolio. Some of the highlights include:

- Secure RHEL and SLES patch download from NCC
- Push or pull based patching.
- Patch rollback.
- Group patching.
- Policy management.
- Server inventory.
- Basic roll based management.
- Requires additional license to be purchased.

The figure below depicts what a base ZENworks Linux Management (ZLM) environment might look like.

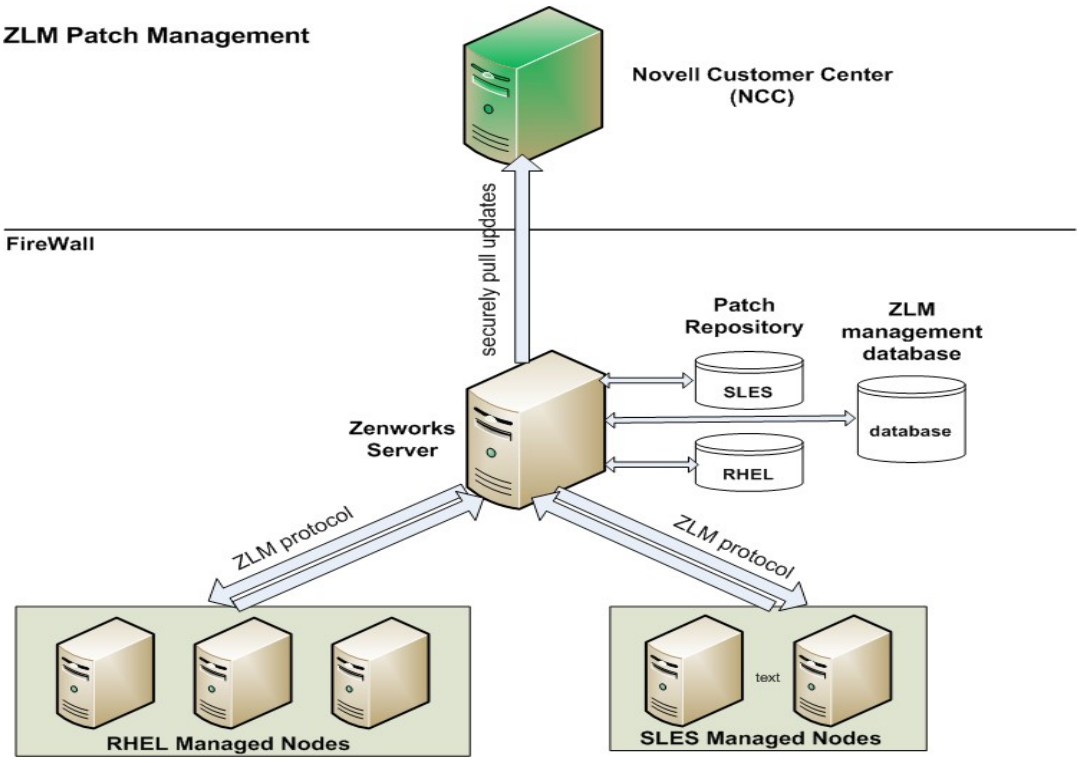

Figure 4: ZLM Patch Management

4.3.3 Integration with Existing Patch Management Tools

It is also possible to integrate with other patching and management solutions such as Opsware and Altiris. In these cases, special connectors are necessary to communicate with Novell's Customer Center.

4.4 Novell SLES Migration Build QuickStart

In order to facilitate a rapid patch management environment for a SLES migration, Novell has developed a process to quickly establish a patch management environment. This process has everything necessary to get a patch management system configured for patching (both RHEL and SLES) via Novell's patches.

5 Application Portfolio

5.1 Application Inventory

In order to have a successful migration you first need to have the best possible view of your current application inventory. Once this inventory is compiled the application classification methods outlined in this section can be used to produce a streamlined approach to migrating your Linux applications.

Novell can assist with creating an application inventory suitable for a RHEL to SLES migration by:

1. Pulling data from pre-existing inventory system(s).
2. Leveraging Novell's own inventory tools.
3. Business and developer interviews.

5.2 Application Classification

Most applications will fall into one or more of the following categories:

1. Distribution services (http, email, nfs, print, etc...).
2. In-house scripted code (perl, python, ruby, etc...).
3. In-house managed code (java, mono, etc...).
4. In-house statically compiled, no external dependencies.
5. In-house statically compiled, with external dependencies.
6. In-house shared libraries, no external dependencies.
7. In-house shared libraries, with external dependencies.
8. ISV software, no internal code components.
9. ISV software, with internal code components.
10. [sub classification on whether “In-house” source code is still available or only available in binaries]

5.3 Application Difficulty Rating

In addition to application classification, Novell has developed a rating system to help properly align resources to each application migration. The rating boils down to a 1 to 5, where 1 is the easiest and 5 is the most difficult.

Rating System:

1. Application works without any intervention.
2. Application works with minor intervention.
3. Application requires a recompile with all dependencies being met.
4. Application requires a recompile along with dependency resolution.
5. Application does not currently work due to lack of support and/or code.

There is also the situation where applications may not be worth the migration investment. These applications generally fall into the following scenarios:

1. Application is in the process of being re-written or on the verge of a major release update.
2. Application ISV dependency no longer exists.
3. Application source code not available.

In these situations it may make the most sense to either black box the application and/or run it as a virtual machine guest on SLES.

5.4 Development Tools and Libraries

A successful RHEL to SLES migration must take into account the current development tools and libraries. In order to facilitate ease of migration the base tools and libraries from the current Operating System release must be met by the new SLES release.

5.5 Novell SLES ISV Matrix

To quickly identify SLES ISV support, Novell has developed an ISV support matrix. This matrix has over 2000 ISV entries for SLES10 and can be leveraged to determine whether a particular third party RHEL application will run and be supported on SLES.

5.6 Regression Testing

An application migration from RHEL to SLES must have a complete feature set and perform at least as well as it did before the migration**. The migration must also not introduce any new bugs in the software.

** unless prior agreement to note that performance and/or features may not be equivalent.

6 Appendix: Migration Program Tools

6.1 Server Migration Guide

This guide is aimed at a server level migration. It includes information on the most common server configurations and how to best approach a server migration from a RHEL server to a SLES server.

6.2 Services Migration Guides

These guides are designed to assist in the migration of specific services such as NFS, Samba, Apache, BIND, Squid and CUPS.

6.3 Migration Utilities

A collection of utilities designed to help with many aspects of RHEL to SLES migration, include:

- create_install_source
  - A tool to automate the creation of a remote SLE build trees for new installations.
- lsb-app-checker
  - A tool that verifies an applications LSB compliance. It also checks for distribution support and library/tool compatibilities.
- AutoYaST kickstart conversion utility
  - A tool to automatically convert kickstart configuration templates to AutoYaST build templates.

7 Appendix – Related Documents

The following table lists related RHEL to SLES migration documents:

Available RHEL to SLES Migration Documents
RHEL to SLES ***Apache2*** Service Migration
RHEL to SLES ***Bind*** Service Migration
RHEL to SLES ***NFS*** Service Migration
RHEL to SLES ***Samba*** Service Migration
RHEL to SLES ***Squid*** Service Migration
RHEL to SLES ***CUPS*** Service Migration

Table 1: Related RHEL to SLES Migration Documents

Novell®