

# SUSE<sup>®</sup> Linux Enterprise Server Deployment Approach Guide

<b>Table of Contents:</b>	<b>2</b> . . . . . Executive Summary
	<b>3</b> . . . . . Deployment Best Practices
	<b>3.</b> . . . . . Breakdown of Application Types
	<b>5</b> . . . . . Managing a Deployment: Phase 1—Inventory
	<b>7.</b> . . . . . Managing a Deployment: Phase 2—Evaluate
	<b>7.</b> . . . . . Managing a Deployment: Phase 3—Prioritize
	<b>9.</b> . . . . . Managing a Deployment: Phase 4—Delivery
	<b>14</b> . . . . . Testing Best Practices
	<b>14</b> . . . . . Common Stumbling Blocks
	<b>15</b> . . . . . Appendices



# Executive Summary

## Novell Keys to a Successful Migration

- Stay focused on business objectives, and start the planning process early
- Plan at every step
- Achieve key stakeholder buy-in beforehand through clear communication of benefits
- Agree on what constitutes success before starting the project
- Capture baseline performance and then again after the migration to show ROI
- Test, test, test
- Building in parallel (new systems next to the old/pre-migration systems/services) and transition over with load-balancers

This document is intended for IT professionals and managers who are considering deploying SUSE<sup>®</sup> Linux Enterprise Server. Novell has had a number of requests from customers looking for help specifically in moving from Red Hat\* Enterprise Linux to SUSE Linux Enterprise Server. In addition to what customers have told us, we will share insights that Novell has gained, having helped numerous customers transition to the most open, most interoperable and most manageable Linux\* platform. The focus of this document will be on deploying SUSE Linux Enterprise Server in environments that previously ran Red Hat Enterprise Linux. This document will highlight key differences between the two platforms, and outline a general deployment plan. After reading this document, current users of Red Hat Enterprise Linux should understand how to become proficient with SUSE Linux Enterprise Server and what a successful migration entails. An in-depth treatment of SUSE Linux Enterprise Server administration is outside the scope of this document. We encourage users to participate in Novell training modules that explore these topics in depth.

While transitioning from Red Hat Enterprise Linux to SUSE Linux Enterprise Server is a straightforward process, Novell has found that careful planning will ensure a smooth transition. Applications that run on Red Hat Enterprise Linux will often run on SUSE Linux Enterprise Server without any changes, so application portability is typically not an issue. What can be challenging is the transition for a systems administrator who is comfortable with the tools specific to Red Hat Enterprise Linux to those tools included with SUSE Linux Enterprise Server. But, once familiar with SUSE Linux Enterprise tools, administrators will find the transition well worth the effort. Novell provides extensive training and support to make

that transition as simple as possible. An additional challenge comes with certifying that the application is performing as expected once it is deployed on SUSE Linux Enterprise Server. While there is no perfect way to handle this, Novell can recommend a variety of activities to help mitigate any difficulties.

Often, SUSE Linux Enterprise Server deployments will be part of a larger refresh cycle. Novell has experience deploying SUSE Linux Enterprise Server in the most complex environments. No matter how daunting the task may seem, the benefits of deploying SUSE Linux Enterprise Server make the undertaking worthwhile. This document incorporates the knowledge that Novell has gained through many successful deployments and brings transparency to a process that is not often well understood.

The following sections will illustrate the methodology that Novell uses for ensuring that a deployment proceeds without business disruption so that a data center reaches its maximum potential. This document will also highlight the differences between Red Hat Enterprise Linux and SUSE Linux Enterprise Server so an organization can learn what they need to learn. Novell recognizes that deploying a new platform can be a daunting task for an organization that has thousands of servers running a multitude of workloads, which are spread over many business units and geographic locations. Often the most difficult part of crafting a deployment strategy is deciding how to start tackling the problem. While each deployment will follow its own path depending on an organization's business requirements and existing applications and hardware, there are common principles that will apply to every situation. A well planned deployment will allow an organization to

both move its current workloads to SUSE Linux Enterprise Server and easily deploy future workloads.

## Deployment Best Practices

Before considering a deployment, it is essential that an organization understand why it is transitioning its workloads to SUSE Linux Enterprise Server. This is important because it allows for both broad buy-in from stakeholders and increased focus on the business objectives. A common reason for deploying SUSE Linux Enterprise Server is to reduce the cost and complexity of a server environment. Another reason is to take advantage of the only Linux platform recommended by Microsoft. It is the only Linux operating system that is optimized to work seamlessly with your Microsoft\* Windows\* environment.

There are three broad areas to consider when crafting a deployment strategy: how to get old applications converted and running on a new platform; how to build and deploy servers; and finally, how to manage the new servers once they are in place. The following best practices were developed to answer these questions and to give structure to a confusing process.

The scope of a deployment depends heavily on the servers or group of servers being transitioned and the size of the data center. If there are just a few workloads that are targeted for migration from Red Hat Enterprise Linux to SUSE Linux Enterprise Server, then all of the rigorous procedures applied below may not be appropriate. However, even the smallest deployment must be planned for, tested and certified. Novell recommends a phased approach, which allows you to implement small pieces your solution a piece at a time, to ensure success and achievable milestones. Deployments are much more manageable when simple workloads are targeted first, with the more intricate workloads coming after establishing some success.

Red Hat Enterprise Linux to SUSE Linux Enterprise Server migrations typically involve four stages. Each stage must be completed before proceeding to the next.

- **Stage one**—*Full inventory of all hardware and software applications; baseline metrics to measure success against*
- **Stage two**—*Evaluation of inventory and determining workload characteristics*
- **Stage three**—*Prioritization of workloads from stage two and development of a plan that maximizes the return through consolidation*
- **Stage four**—*Delivery of initiatives into production, including rigorous testing, using a methodology that is agreeable to the organization*

It is very important for enterprises to conduct proper testing before deploying new solutions into the production environment. Often the most difficult part of a new deployment is agreeing on the criteria for the new environment to be considered functional, as well as ensuring that all components have been identified and inventoried. We will cover what is required for proper testing so that the transition is seamless.

## Breakdown of Application Types

Novell has found that there are three broad categories of applications that are typically targeted for redeployment. The complexity of migrating the application varies with the type. Below is a brief overview of these categories, and the types of activities involved in migrating these applications.

### First Tier—Roads and Sewers

**Scope:** This involves standard network and other services common across many variants of UNIX\* and Linux, such as Apache, DHCP, sendmail, BIND (DNS), LDAP, NIS, NFS and others. There is also integration with existing “roads and sewers” services, such as backup, monitoring, security, auditing, user management and more.

**There are three broad areas to consider when crafting a deployment strategy:**

- **how to get old applications converted and running on a new platform**
- **how to build and deploy servers**
- **how to manage the new servers once they are in place**

**Activities:**

- *Install package*
- *Apply existing configuration (from old system) to new installation*
- *Test for correct operation*

**Complexity:** Low technical complexity. Standard open source components such as Apache will move over with little or no significant reconfiguration. That is, the configuration for Apache (httpd.conf and its subtrees) should be able to be copied into the new version of SUSE Linux Enterprise Server—along with the actual Web site content—and work with minor or no changes. Boundary services such as DHCP and BIND (DNS) can be migrated in a similar fashion.

**Second Tier—Mid-stack Enablers**

**Scope:** This typically involves large commercial, off-the-shelf (COTS) components such as SAP, Oracle, PeopleSoft, Serviceguard and Veritas.

**Activities:**

- *Verify that the COTS component is certified for SUSE Linux Enterprise Server (check at [www.novell.com/partner/isv/isvcatalog](http://www.novell.com/partner/isv/isvcatalog))*
- *Install base COTS package on SUSE Linux Enterprise Server*
- *Move customizations from old system to new*
- *Test and re-certify correct operation on SUSE Linux Enterprise Server*

**Complexity:** Basic installation is straightforward. Migrating the actual configuration and certifying is typically a more complex activity. This is best done by the resources that performed the original customizations (Accenture, Deloitte & Touche, etc.).

**Third Tier—Custom Line-of-Business Applications**

**Scope:** These are the homegrown applications developed by a client to address its unique business processes and value proposition.

**Activities:**

- *Inventory of all software to be ported*
- *Complete inventory of existing testing assets*
- *Develop customized project plan*
- *Port (with a Linux-to-Linux migration, this step is essentially eliminated)*
- *Unit test*
- *Integration test*
- *System test*
- *User acceptance test*

**Complexity:** Technically not complex because of the inherent compatibility of Linux. Testing brings additional complexity. However, this is also the area that typically sees the highest payback. Within this category, software can generally be subdivided further into several buckets with varying levels of complexity:

- *In-house statically compiled, no external dependencies*
- *In-house statically compiled, with external dependencies*
- *In-house shared libraries, no external dependencies*
- *In-house shared libraries, with external dependencies*

*Sub classification on whether in-house source code is still available or only binaries categories*

**Key Differences between Red Hat Enterprise Linux and SUSE Linux Enterprise Server**

While many aspects of Linux are similar from distribution to distribution, there are a few key differences between Red Hat Enterprise Linux and SUSE Linux Enterprise Server. These areas are all covered in the Red Hat Enterprise Linux to SUSE Linux Enterprise Server migration training modules that can be obtained from Novell.

### AutoYaST vs. Kickstart

Red Hat Enterprise Linux includes a program called Kickstart, which is used to save and deploy server configurations. SUSE Linux Enterprise Server employs a powerful tool called AutoYaST that delivers the same functionality. While the end result is similar, Novell recommends that administrators be retrained to use AutoYaST.

### AppArmor® vs. SELinux

Red Hat Enterprise Linux administrators who are familiar with SELinux will be happy to know that AppArmor achieves the same level of security with far less complexity. SELinux rules will have to be translated to be used with AppArmor. However, AppArmor is much easier to manage and includes many ready-to-use rulesets for the most common services.

### Directory Structure

SUSE Linux Enterprise Server and Red Hat Enterprise Linux differ on some aspects of their directory structures. Some configuration files may not be located exactly where a Red Hat Enterprise Linux administrator would expect them. The SUSE Linux Enterprise Server documentation is very helpful in determining the location of each configuration file. In terms of philosophy, SUSE Linux Enterprise Server is closer to and adheres to the LSB/FSH and Red Hat Enterprise Linux is more concerned with backward compatibility of Red Hat Enterprise Linux. Solaris administrators will easily become familiar with SUSE Linux Enterprise Server.

### Basic Commands

There are some basic configuration commands that differ between SUSE Linux Enterprise Server and Red Hat Enterprise Linux. Instead of using `system-config-*` or `redhat-config-*`, SUSE Linux Enterprise Server employs the use of `yast2*` for certain configuration activities. For a full list and description of these commands, please refer to the Red Hat to SUSE Linux Enterprise Server training module, which is free from Novell.

There is an additional document that is part of the deployment kit, available from Novell, which covers this topic in more depth.

### Clustering

While both distributions use ext3 as the default file system, they differ on their clustering implementations. For clustering, Red Hat Enterprise Linux relies on the Red Hat Cluster Suite which uses the Global File System (GFS), while SUSE Linux Enterprise Server uses OCFS2 with Heartbeat. Administrators who are familiar with Red Hat Cluster Suite will need training on the clustering features in SUSE Linux Enterprise Server.

### Management

Red Hat Enterprise Linux administrators should be familiar with using Red Hat Satellite Network to administer their UNIX machine. SUSE Linux Enterprise Server installations are typically managed using Novell® ZENworks® Linux Management or the Novell System Management Tool (SMT), so staff will have to be trained to use it.

## Managing a Deployment: Phase 1—Inventory

The inventory phase provides a starting point for a new deployment. A thorough inventory is critical to the success of the later steps. The inventory phase can be broken into five steps: hardware, OS, storage, stack and function. For each step, the following details must be collected:

### Hardware

- *Manufacturer*
- *Architecture (32/64 bit)*
- *CPU type (e.g., x86, AMD64, EM64T, IA64, PPC, PA-RISC)*
- *Number of CPUs*
- *Number of cores*
- *RAM*
- *Firmware versions*
- *Projected hardware refresh date/EOL*
- *Certifications/support levels*

## Operating System Type and Version

- *Type (Solaris\*, Red Hat Enterprise Linux, AIX\*, etc.)*
- *Version*
- *Release level (service pack)*
- *Architecture (32/64 bit)*
- *Certifications/support levels*

## Storage Requirements

- *How storage is currently configured*
- *How storage will change*
- *Local/NAS/SAN*
- *HBA/Multipath requirements*
- *Firmware requirements*

## Stack

The stack includes all software in a standard build, with the exception of the line-of-business applications that the server hosts. That includes, but is not limited to security, backup, management and monitoring software. You must take care to ensure that the information captured will allow you to rebuild a similar stack on a SUSE Linux Enterprise platform. The information gathered here will serve as a baseline for delivering the application on a test platform.

Important information includes, but is not limited to:

- *Running processes*
- *Libraries*
- *Package versions*
- *Applications*
  - *Package dependencies (e.g., other applications, processes, systems, userids, etc.)*
  - *Software vendor*
  - *Independent software vendor (ISV) support level*
  - *Version*
- *Connectivity to external storage devices*
- *Security restrictions*
- *Authentication methods*

## Function

The purpose here is to capture as much information as possible about the applications to be migrated. They can be broken down into several subcategories:

### Enterprise Database Server

Information about the database server and where data is stored must be captured. Whether or not to migrate the database server is an important consideration; most of the popular databases are supported on SUSE Linux Enterprise Server today.

### Infrastructure and Edge Server

This includes applications such as VPN, firewall and directory/print. Relevant information includes:

- *Service type*
- *Number of users*
- *Data requirements*
- *Client connectivity methods*
- *Configuration files*

### ISV Business Applications

This is perhaps the most important category; it is necessary to ensure that these applications have been certified to run on SUSE Linux Enterprise Server. It is important to map out all third-party dependencies to ensure that these are available as well. An ISV will often certify its offering on SUSE Linux Enterprise Server if prompted to do so by a customer, so current support is not necessarily a road-block to migration.

### Custom Applications

Custom applications are any applications developed in house that will require porting and testing. The effort involved in porting applications is heavily dependent on the code itself. There are many open source and commercial tools available to assist with assessing application portability. In general, applications that sit on top of middleware (e.g., Oracle Forms on Oracle\*, WebLogic\*,

WebSphere\*, J2EE\*/Tomcat) are the easiest to port. Applications developed using scripting languages are typically straightforward as well, though testing is required before deployment.

### **Performance Baselines/Metrics**

Baseline metrics can both promote what was gained and establish success by measuring performance before the migration. There is a need to measure performance of key systems (sustained and peak loads, etc.) and standard operations (new record adds, batch processes, backups, other regular operations) and establish baselines. Without baselines it is difficult to document how the migration was a success.

## **Managing a Deployment: Phase 2—Evaluate**

The evaluation phase involves a careful analysis of the data gathered in the inventory phase. This entails a close look at the data collected from the inventory, as well as additional information gathering that can be used to prioritize which workloads to deploy first. The analysis can be broken down into the following steps:

### **Hardware Review**

The first consideration is what to do with the existing hardware. There are several options, including using existing hardware to support SUSE Linux Enterprise Server and retiring current hardware if it does not meet future requirements.

### **Application Lifecycle**

To determine which applications to target first, it is important to consider where each application is in its lifecycle. Applications nearing the end of their expected lives are excellent candidates for redeployment.

## **ISV Certification Status on SUSE Linux Enterprise Server**

For each ISV or COTS application, investigate its current support level on SUSE Linux Enterprise Server. Important considerations include whether the application runs on SUSE Linux Enterprise Server, is supported on SUSE Linux Enterprise Server, has a newer version available for SUSE Linux Enterprise Server, or has any dependencies that are not available. If the application is not currently supported, consider the ISV's roadmap, or whether another application that is supported can replace the function. For more information on ISVs and independent hardware vendors (IHVs) supported on SUSE Linux Enterprise Server, visit: [www.novell.com/partner/isv/isvcatalog](http://www.novell.com/partner/isv/isvcatalog)

It is well documented that most ISVs will respond to customer requests for OS support. Customers should make their intentions explicit with ISVs to obtain certification for applications they want to migrate.

### **Profiling**

Profiling is critical to realizing the benefits associated with deploying SUSE Linux Enterprise Server. To plan for consolidation, it is extremely important to understand the characteristics of each workload. Workloads vary in their characteristics over time, so it is important to profile workloads over days, weeks and months before migration.

## **Managing a Deployment: Phase 3—Prioritize**

The next task is to discover which hardware and/or applications are prime targets for redeployment. The end result should be a roadmap that the organization can use to plan for current deployments and future application delivery.



## Criteria and Weighting

We apply three criteria to create a roadmap of activities that will deliver maximum benefit and impact.

- **Risk (High/Medium/Low)**—*The business risk of making this change outside of normal operational ever-greening policies*
- **Complexity (High/Medium/Low)**—*The technical complexity of making the change; e.g., migrating network infrastructure services is a less complex task than migrating custom-developed applications written using native code. The complexity can generally be judged on a scale from one to five, where one is the easiest and five is the hardest*
  - The application works without any intervention
  - The application works with minor intervention
  - The application requires a recompile with all dependencies being met
  - The application requires a recompile along with dependency resolution
  - The application does not currently work due to lack of support and/or code
- **Opportunity (High/Medium/Low)**—*When viewed within the context of the objectives of migrating to SUSE Linux Enterprise Server, what opportunity does this profile of server or application present? Is the application worth the migration investment? These applications can be grouped into three buckets:*
  - The application is in the process of being re-written or on the verge of a major release update
  - The application ISV dependency no longer exists
  - The application source code not available

In some situations it may make sense to black box the application and/or run it within a virtual machine guest on SUSE Linux Enterprise Server.

The outputs of the prioritization phase should provide the following information:

- Which applications should be deployed first
- Number of hardware build variations for delivery
- Number of software stack variations for delivery
- Opportunities for consolidation
- Opportunities for virtualization

Once these criteria are applied to the matrix, those applications that benefit the most from being redeployed will have a higher priority. These are the applications that will perform better when redeployed on SUSE Linux Enterprise Server, are not difficult to implement, and will have the highest impact on business goals.

## Simple Decision Criteria for Choosing Migration Targets

When conducting an assessment and inventory of servers, consider the following questions:

- Are there numerous similar servers to be redeployed or is it a one-of-a-kind server?
- Are the target business applications supported or certified for SUSE Linux Enterprise Server by the ISV? ([www.novell.com/partner/isv/isvcatalog](http://www.novell.com/partner/isv/isvcatalog))
- Is the target hardware platform supported by the IHV?
- Does the migration allow you to discontinue support and retire costs on the old platform?
- Is the workload one that can be virtualized?
- Is the workload a candidate for running in an Integrated Facility for Linux (IFL) on a

*zSeries\* mainframe (such as a WebSphere application server)?*

- *Would the workload benefit from SUSE Linux Enterprise Server Real Time functionality?*
- *Is this a Windows workload that could be virtualized on SUSE Linux Enterprise Server?*
- *Are there operating system parity issues for critical functions?*

### **Tying Deployment to Upgrade Cycles**

Where possible, SUSE Linux Enterprise Server deployments should be tied to hardware refresh and software upgrade cycles. This minimizes migration costs and maximizes efficiency by creating synergy with existing corporate or line-of-business IT initiatives.

### **Road Map Creation**

The prioritization phase should also include a high-level roadmap creation, implementation planning and scheduling, and migration resourcing. Important quarterly and monthly deployment milestones should be noted in the roadmap. Server deployment targets should be estimated and tracked as part of monthly milestone reporting. A high-level deployment approach is shown depicted below:

### **Deployment Workshops and Deployment Planning**

These short, facilitated whiteboarding sessions help develop the deployment approach and momentum in your organization. They are the primary tool for pulling together the information in the evaluation and prioritization activities. They are typically one day in length, staffed by one senior facilitator and one technical specialist or architect, with a high-level read-out deliverable. These workshops assume active participation of vertical and technical experts. Staffing and duration can be modified or tailored to meet the customer's needs. Deployment workshops, regardless of duration, will typically follow this outline:

- *Review of the business and technical drivers for a solution based on SUSE Linux Enterprise Server; business value justification*
- *High-level application migration and conversion inventory with initial ISV certification requirements*
- *Initial "deep dive" into a single focus area (core build engineering, virtualization, mixed-source systems management, identity interop needs, etc.)*
- *High-level support and education needs assessment*
- *High-level deployment roadmap of enablement services, approach for maximizing certificate usage; and next steps*
- *Executive readout and presentation delivery*

### **Moving from Traditional Stacks to Virtual Stacks**

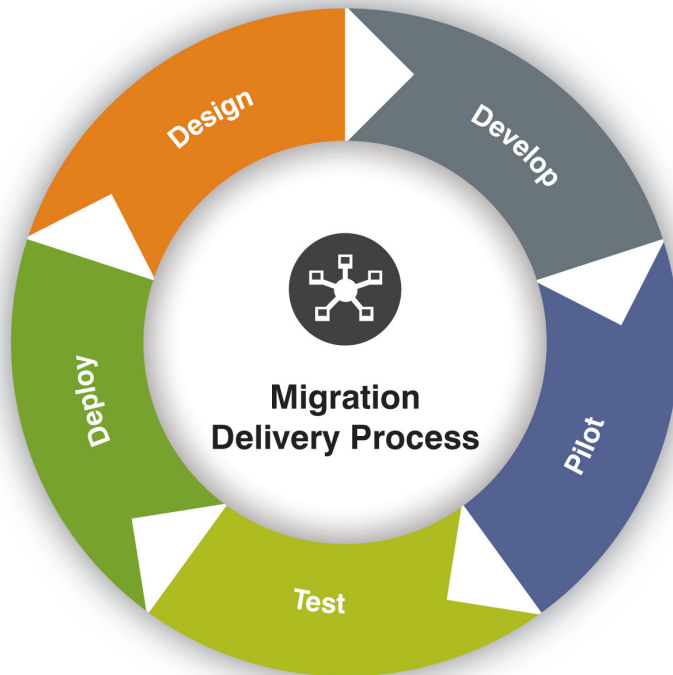
Any migration must consider a virtualization strategy to reap the benefits of modern hardware. Provisioning servers correctly for virtualization is a complex endeavor; however, there are numerous software tools to assist in this task. Novell recommends consulting a third party with prior experience in virtualization as part of the migration process. Virtualization will also provide a method for deploying services that may be extremely difficult to move off of legacy platforms.

### **Managing a Deployment: Phase 4—Delivery**

During the delivery phase, an organization should use the results from the evaluation and prioritization phases to undertake detailed scoping and execution activities. These should translate into a concrete set of deliverables with quantifiable objectives.

The objectives should reflect the business needs for the deployment as well as the

required application and hardware stacks that have been identified.



### Scoping

Determine which applications and hardware models to target for deployment based on the outcomes of the prioritization phase:

- *Define the scope of first phase of deployment and align with objectives*
- *Make sure the scope is approved by project sponsor and key stakeholders*
- *Determine if there are any new requirements*
- *Determine what packages and applications should be included in standard build*
- *Determine how many builds will be required to support required software stack/hardware variants*
- *Prepare a project plan of the tasks required to achieve each deliverable*
- *Align personnel to delivery tasks*
- *Scope the server and supporting infrastructure requirements for pilot and production*
- *Begin hardware and software procurement process*

### Design

Prepare a design for the core application and hardware builds detailing:

- *Make/model/architecture of target hardware*
- *Core SUSE Linux Enterprise Server packages to be included in build*
- *What core build sources will be required*
- *What additional application sources will be required*
- *What application dependency sources will be required*
- *Patching requirements*
- *Security/hardening requirements*
- *Preparation of use case scenarios for testing stage*
- *Documentation requirements*
- *Think about how the delivery cycle could be integrated into the organization's existing hardware, and software lifecycles for patching, upgrade, refresh and procurement.*

### Configuration

The configuration stage refers to the preparation of the test and production environments in readiness for the preparation of core build and application porting activities. After the configuration phase, an organization will have a dedicated lab environment from which the rest of the activities originate. There are three key steps in configuration, and Novell provides extensive documentation to assist with each step:

- *Prepare the hardware for the lab*
  - Procure and configure hardware for the installation server
  - Procure and configure hardware for the development environment
  - Procure and configure hardware for testing builds
- *Preparing the software for the lab*
  - Create SUSE Linux Enterprise Server install sources for each architecture (e.g., i386, x86-64)
  - Create RPM packages for software if they do not exist

- Create signed add-on repositories for application packages and dependencies which aren't available as part of the core SUSE Linux Enterprise Server core distribution
- Create installation sources for add-on applications and packages
- *Prepare the build delivery mechanism*
  - Identify required build types (e.g., network, DVD, USB)
  - Decide on boot process for connecting to builds (e.g., PXE, CD, DVD)
  - Decide on the protocol or protocols to be used for build delivery (e.g., HTTP, FTP)
  - Prepare the installation server services for delivery methods
  - Build multiple delivery methods from the same source

Novell recommends setting up the environment to allow a PXE boot that pulls a XML control file from an NFS repository. Organizations that do not wish to modify their DHCP server may want to use HTTP or FTP instead. We urge organizations to consider using SSH to deliver images as they may contain sensitive data.

Organizations should try to make their lab environments as close as possible to the production environment because issues can arise if they are not closely matched. The lab environment should be maintained after the migration project in order to stage new applications into the production environment.

## Implementation

The implementation phase includes creating the core builds for the required stack and the development activities related to application porting.

### Creating the Core Build

AutoYaST is the tool bundled with SUSE Linux Enterprise Server that allows a system administrator to capture a server's hardware

and application build parameters and store them in an XML profile. The profile that is created can then be used to recreate servers with the same hardware configuration and software stack. The XML can be used as a baseline and updated as required.

In order to create the core build, first create a reference build on the target hardware and create a copy of the AutoYaST XML file. Edit this XML file as needed for each of the other desired builds. It is important to extensively test the reference build before using it to create the other core builds. Reference the AutoYaST documentation for more information on customizing the XML file.

### Build in Security

- *Configure required network firewall rules with the kernel iptable packet-filtering module.*
- *Follow best practices for securing SUSE Linux Enterprise Server*
- *Create AppArmor profiles for hosted applications*
- *Consider using an intrusion detection system (IDS)*
- *If not already included in the application stack consider monitoring software, e.g., Nagios or BMC PATROL*

### Test the Build

As part of the testing, make sure that:

- *The servers build without error*
- *The servers meet all requirements*
- *System security meets organizational requirements*
- *Performance is as expected*
- *Typical server loads are easily handled*

If there is an internal security team, make sure that they approve the build and include any input they may have. Any bugs found should be fixed, and unit testing should be completed before any new build is deployed into the production environment.

Novell encourages organizations to develop a formal process for creating new builds. No rogue builds should exist in the production environment. Organizations are encouraged to always follow a formal “design, build, test, deploy” cycle when deploying any new hardware or software.

### **Document the Core Image Build Process**

All the steps taken to create the core image should be carefully documented so they can be reproduced. Information should be included about:

- *Building the installation server*
- *Using and maintaining the master installation server*
- *Creating and using control files*
- *Setting up network booting*
- *Deploying the build in a test environment*

### **Conduct a Production Pilot**

Once a build has been successfully deployed and tested in the lab, it can be introduced into a production environment. Novell highly recommends that organizations limit initial deployment to a select group of users. End-to-end system tests and workload evaluations should be completed as part of the pilot.

### **Roll Out**

Once the pilot is successfully completed, the full rollout can begin. It is extremely important to communicate with users and quickly identify and fix any issues as they arise. The lab environment can never completely emulate the test environment, but careful planning to this point will serve to minimize disruption.

### **Maintenance**

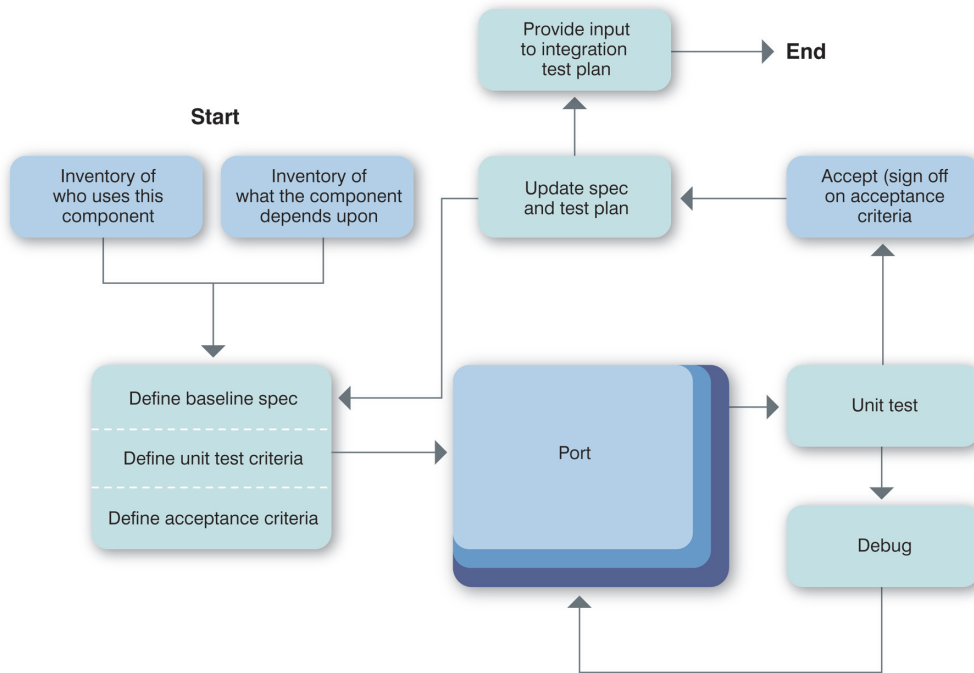
As part of the maintenance phase, consider how the servers will be managed once they are rolled out. Novell supports three methods for managing patching and distribution:

- *Novell ZENworks Linux Management for complete Linux patch management and software distribution*
- *Subscription Management Tool can securely pull Novell patches from behind a firewall and can be used for basic patching functionality when combined with yum or AutoYaST*
- *Integration with existing patch management tools is possible in some situations. In these cases, a special connector is required for third-party tools to communicate with Novell Customer Center*

### Application Porting Process

Novell recommends following the process in the following diagram:

*This is the recommended Novell approach, but even some subset would help avoid problems; lack of a formal approach limits cost reduction opportunities.*



### Linux Porting

Applications generally do not have to be ported from Red Hat Enterprise Linux to work on SUSE Linux Enterprise Server. At most, the location of certain settings may have to move. The most important consideration is the set of libraries that were installed on the old system, and ensuring that the libraries used on the build of SUSE Linux Enterprise Server are compatible. Consult the ISV supporting the application to learn if any special steps are required.

For applications developed in house, it may be possible to perform a standard port from an older distribution of Linux to SUSE Linux

Enterprise Server. Novell and other companies such as IBM, Hewlett Packard and Intel have produced many documents, guides and supporting software to allow an application to be migrated, either directly or via an intermediate step.

Applications being examined for porting should be appraised on the basis of their conformity to the Linux Standards Base (LSB). The LSB is a standard developed by The Linux Foundation ([www.linux-foundation.org/en/LSB](http://www.linux-foundation.org/en/LSB)) that maps the libraries, commands and utilities, file system structure and other system characteristics required for a compliant Linux distribution to provide support for applications.

SUSE Linux Enterprise Server 10 supports LSB 3 (LSB 1.3). Applications that have been developed to support an earlier version of the LSB should be prioritized for porting to SUSE Linux Enterprise Server to accelerate porting time and reduce the effort required. Applications written to conform to LSB 2 (LSB 1.2) standards, such as those deployed on Red Hat Enterprise 5 servers, should be a higher priority for porting over applications that do not conform to any LSB standard. Part of the application inventory process should be geared towards identifying applications that have a dependency on the LSB package, a process that can be performed by standard Linux package management system tools. Analysis of the application can be performed using tools and utilities that are supplied with the LSB Development Kit (LDK). These utilities can confirm to the developer the steps required to make the application LSB compliant.

### **Subscription Management**

Processes should be put in place for managing the following:

- *Subscription entitlements allocated or designated to particular lines of business*
- *Users that are allowed rights to Novell Customer Center*
- *Subscription server installation of usage*

### **Testing Best Practices**

Novell has learned that one of the most critical parts of any deployment is determining what constitutes success. That is, there must be agreement among all parties about the specific functionality that the new system must meet in order for it to be deemed complete. Testing requirements vary significant from organization to organization. As part of the deployment strategy, it is important to determine the existing assets for testing, and to develop a plan to create new testing assets to satisfy all requirements. Some basic things to remember when testing include:

- *Inventory existing test assets in the discovery and planning stage*
- *Define additional test assets required for deployment*
- *Get stakeholder on acceptance process and criteria prior to deployment*
- *Testing is a continuous activity from beginning to final acceptance of redeployed system; it is interwoven with all deployment tasks*

### **Common Stumbling Blocks**

Large data centers often do not have a reliable inventory of hardware and software in production. That is, they cannot easily answer the question: Exactly what needs to be migrated, and on which machines? This risk can be mitigated by spending ample time in the inventory phase. It is not enough to simply write down a list of all the servers; information must be collected about the applications that are running and all of their dependencies. Before leaving the inventory phase, an organization should know not only what is running in their data center, but also how everything interacts. The best way to achieve a complete inventory is to follow the protocol described in the inventory section of this paper.

Many IT organizations do not have formal test, acceptance and certification criteria and/or test harnesses for their stacks. That is, they cannot easily answer the question: How can I tell if the migrated stack is working properly? It is important to carefully determine the criteria for success before beginning the migration. If those criteria are not currently in place, they must be developed. Both this, and the inventory issues must be identified and dealt with very early in the process. Novell recommends that organizations that do not have inventory or testing procedures in place contact a third-party firm that has experience working in such environments.

Many organizations have IT departments that have followed a path of irregular, organic growth. In these cases the applications running in the data center were added as needed, without any formal and repeatable project management in place. The end result is a mixture of systems, platforms and applications that vary in complexity. The IT landscape is not well understood by a single person or group of people. That means there is nobody who can easily answer the question: How well is my migration going? Novell recommends establishing a formal program management office for the duration of the migration project, and beyond, if one does not already exist. This office can be as small as a single person, but at least one person must have primary responsibility for tracking the progress of the migration by mapping it to specific deliverables.

Deployment projects often touch some mission-critical aspect of an organization's business. As a result, operational teams often adopt the mantra of "don't touch it if it isn't broken" and tend to resist change. It is therefore essential that a migration project have the sponsorship of a senior executive sponsor within the organization. While repeated communication of the benefits of deployment will help overcome objections, an internal, influential worker is needed to achieve success. A strong argument for deployment typically centers around the reduction in hardware and software licensing costs that will be achieved by reducing vendor lock-in. Specific benefits can usually be found to appease individuals who still raise objections.

## Appendices

### Further Reading

The content of this document are intended to give a high level overview of the best practices for starting a project for migrating from Red Hat Enterprise Linux to SUSE Linux Enterprise Server.

The links below provide will allow readers to explore the mentioned migration processes in more detail.

### Application Compatibility

SUSE Linux Enterprise Software Catalog  
[www.novell.com/partner/isv/isvcatalog](http://www.novell.com/partner/isv/isvcatalog)

Migrating to SUSE Linux Enterprise  
Novell Course 3068 Migrating from  
Red Hat to SUSE Linux Enterprise Server 10  
<http://download.novell.com/protected/Summary.jsp?buildid=ReHgN8b1qhU~>

### Core Build

AutoYaST Documentation  
[www.suse.com/~ug/autoyast\\_doc/index.html](http://www.suse.com/~ug/autoyast_doc/index.html)

AutoYaST FAQ  
[www.suse.com/~ug/AutoYaST\\_FAQ.html](http://www.suse.com/~ug/AutoYaST_FAQ.html)

AutoYaST Mailing List  
<http://lists.opensuse.org/opensuse-autoinstall/>

SUSE Linux Enterprise 10 AutoYaST  
<http://forgeftp.novell.com/yast/doc/SL10.0/autoinstall/index.html>

Novell Forge: Importing Red Hat  
Kickstart Profiles into AutoYaST  
<http://forgeftp.novell.com/yast/doc/SL10.0/autoinstall/importkickstart.html>

### Security

Novell: Linux Security  
[www.novell.com/linux/security/](http://www.novell.com/linux/security/)

Novell: Application Security with AppArmor  
[www.novell.com/documentation/apparmor/](http://www.novell.com/documentation/apparmor/)

Securing Your Applications: Deploying Novell  
AppArmor with ZENworks Linux Management  
[www.novell.com/connectionmagazine/2006/q3/pdf/techtalk4\\_q306.pdf](http://www.novell.com/connectionmagazine/2006/q3/pdf/techtalk4_q306.pdf)

Center for Internet Security: SUSE Linux



Enterprise Server Security Benchmark  
[www.cisecurity.org/tools2/linux/CIS\\_SUSE\\_Linux\\_Benchmark\\_v1.0.pdf](http://www.cisecurity.org/tools2/linux/CIS_SUSE_Linux_Benchmark_v1.0.pdf)

**Data Center Automation**  
Novell ZENworks Orchestrator  
[www.novell.com/products/zenworks/orchestrator/](http://www.novell.com/products/zenworks/orchestrator/)

[www.novell.com](http://www.novell.com)



Contact your local Novell  
Solutions Provider, or call  
Novell at:

1 800 714 3400 U.S./Canada  
1 801 861 1349 Worldwide  
1 801 861 8473 Facsimile

**Novell, Inc.**  
404 Wyman Street  
Waltham, MA 02451 USA