Open Enterprise Server 2015 SP1
Web Services and Applications Guide

June2016

Novell

Legal Notices

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see https://www.novell.com/company/legal/.

Contents

About This Guide

Open Enterprise Server (OES) 2015 SP1 includes a collection of open source web services and applications that let you build, deploy, host, and use websites and web applications that speed up business processes without jeopardizing the security of business information. This guide introduces you to some key web services and applications, and explains how to install them on OES 2015 SP1 servers.

The guide is divided into the following sections:

Audience

This guide is intended for web or network administrators that install and manage website content and applications. Developers that write web-based applications to run in the OES environment might also find the information in this guide helpful.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation.

Documentation Updates

For the most recent documentation, visit the OES 2015 Documentation website.

Additional Documentation

Each open source software component discussed in this overview has its own documentation on the web. For details about how to configure and manage each component, refer to the following documentation:

- Apache 2.2 documentation (http://httpd.apache.org/docs/2.2/)
- Tomcat 6 documentation (http://tomcat.apache.org/tomcat-6.0-doc/index.html)
- MySQL documentation (http://dev.mysql.com/doc/)
- PostgreSQL documentation (http://www.postgresql.org/docs/)
- PHP Hypertext Preprocessor documentation (http://www.php.net/docs.php)
- Perl documentation (http://www.perl.org/docs.html)
- Python documentation (http://www.python.org/doc/)
- Ruby on Rails documentation (http://guides.rubyonrails.org/)

1 Overview of Web Services and Applications

Open Enterprise Server (OES) 2015 SP1 includes a collection of open source web application services products that let you build, deploy, host, and use websites and web applications that speed up business processes without jeopardizing the security of business information.

IMPORTANT: The following open source software packages require additional product-specific support contracts to be obtained by the customer in order to receive full support:

- ◆ MySQL Database
- ◆ PostgreSQL Database
- ◆ WebSphere Application Server Community Edition (This package is not available in SUSE Linux Enterprise Server 11 SP3 and later.)

Documentation and community support are available from the open source communities.

This section covers the following topics:

1.1 Introduction to Web Services and Applications

The rise of the Internet and the World Wide web sparked a revolution not only in network communications but also in application design and development. Programmers have encapsulated pieces of business functionality into distinct objects or components, and then made them available as self-contained web services that can be accessed using Internet-based protocols and tools.

As network servers have become capable of supporting Internet-based services, software developers have devised new programming paradigms to take advantage of the widespread availability of these services. This new class of software is categorized as web-based or web-enabled applications.

This section introduces some basic concepts and technologies that are helpful to understand when working with web application services.

1.1.1   What Are Web Services?

The term *web services* can be confusing because it is used in many different ways. In most contexts, web services are business logic components that can be connected together and exchange data to perform a useful task. The components can be internal or external to an organization, and they communicate using Internet-based protocols such as the HyperText Transfer Protocol (HTTP). In brief, web services run on servers and process substantial amounts of data that users want to be able to access quickly and easily.

A popular programming model in which individual web services are combined to create a functional whole is the *service-oriented architecture*. In this model, a service consumer sends requests to a service provider over a standard connection. The request and subsequent response are defined in a way that is understandable to both the consumer and provider.

Most web services use Extensible Markup Language (XML) to define the format of request and response messages. XML features a tagged structure that provides the needed flexibility for exchanging data between disparate components. XML can also be used to define how data is stored in a database.

Simple Object Access Protocol (SOAP) provides a standard for enveloping and sending web services messages. It is an XML messaging specification that describes a message format along with rules for exchanging data in the proper sequence between structured data types and arrays. SOAP generally uses HTTP, but it can use other standard web protocols as well.

In the service-oriented architecture, service consumers can find available service providers through various discovery mechanisms. One such mechanism is the Universal Description, Discovery, and Integration (UDDI) registry. As web services are developed, they can be added to the UDDI registry. The registry can then be searched in various ways to find the web services available for a particular organization and obtain contact information.

1.1.2   What Are Web Applications?

In its simplest form, a web application is an interactive system that allows its users to execute business logic that resides on a server and to view the results of that logic through a web browser on a client workstation. The defining factor that makes the system a web application is that the server and client communicate over the Internet. In brief, web applications make the data processed by web services available to users quickly and easily through their web browsers.

Web applications are built on a client/server architecture. The business logic is contained in the application itself, which runs on a web server and uses HTTP to communicate with clients over the Internet. The web server manages the application, passes requests from clients to the application, and returns the application's responses to the client.

On the client side, the web application is viewed with a browser. The application's user interface takes the form of HyperText Markup Language (HTML) pages that are interpreted and displayed by the browser. In addition to text, these HTML pages can contain web forms, image files, audio and video clips, and other types of displayable data.

Although web applications can use a website as the front end to their business logic, you can do many things in a web application that you can't do with a static website, such as:

- Identify specific users and present a customized interface for each user
- Collect information from users and store that information on the server
- Perform tasks for users, such as retrieving information from a database, registering to access specific content, or placing an order for a product

1.1.3   Web Application Tools (Java and J2EE)

Java has become a standard programming language for web applications because it is simple and portable to various hardware platforms. All you need to run Java applications is a Java Virtual Machine (JVM) for your particular platform. JVMs are available for almost every server platform in existence, including SUSE Linux Enterprise Server, Novell NetWare, Sun Solaris, Microsoft Windows, and Apple Macintosh OS.

Java 2 Platform, Enterprise Edition (J2EE) is a widely used environment for developing enterprise web applications. J2EE offers a multitiered distributed application model, the ability to reuse components, integrated XML-based data interchange, a unified security model, and flexible transaction control. Best of all, applications developed for a J2EE application server are not tied to any one vendor's products or APIs.

The J2EE specification defines the following components:

 - ◆ **Servlets:** A Java servlet is a server-side component that provides a simple, consistent mechanism for extending the functionality of a web server and for accessing existing business systems. A servlet dynamically processes client requests and constructs responses. Servlets are commonly used to process forms, handle redirects or authenticate user names and passwords, and create dynamic content for a web application.
 - ◆ **JavaServer Pages:** JavaServer Pages (JSPs) are text-based documents that execute as servlets but allow a more natural approach to creating web content. JSPs allow web developers to rapidly develop and easily maintain dynamic web pages that leverage existing business systems. JSP technology separates the user interface from content generation, enabling the overall page layout to be changed without altering the underlying dynamic content.
 - ◆ **Enterprise JavaBeans:** Enterprise JavaBeans (EJBs) are the basic components of an architecture that allows developers to create objects that precisely model the structure and logic of a business application domain. The system-level details of building the distributed application are abstracted out, enabling domain experts to be developers who freely focus on solving business problems. EJB technology enables rapid development of distributed, transactional, secure, and portable Java-based applications.

1.1.4   Enabling Technologies

Web applications employ various enabling technologies to make their content dynamic and to create user interfaces into the business logic on the server.

Scripting Languages

Foremost among the enabling technologies are scripting languages such as PHP and Perl.

PHP (PHP: Hypertext Preprocessor) is a powerful server-side scripting language that is easy to learn. It offers all of the power and flexibility of JSP, but does not require as much memory and processing power. You mix specially delimited PHP code in with regular HTML to create a dynamic web page. PHP is commonly used to access web databases such as MySQL. It also supports library extensions to leverage standard services such as LDAP, FTP, POP3, Java, and many others.

Perl (Practical Extraction and Report Language) is another server-side scripting language commonly used by web programmers to create scripts for web servers. It uses a syntax similar to C/C++ and its file-manipulation and text-manipulation facilities make it ideal for tasks involving software tools, database access, graphical programming, networking, and system management.

Servlet Containers

A complementary component for both servlets and JSPs is the *servlet container*. The container acts as a simple application server that executes Java servlets and renders web pages that include JSP code. It provides necessary functions such as life cycle management and interaction with a web server.

The official reference implementation of the Java servlet API is Jakarta-Tomcat, an open source project released under the Apache Software Foundation. Tomcat is typically used in conjunction with a web server such as Apache.

Web Database Servers

MySQL is an open source, structured query language (SQL) web database server that is often used by PHP and Perl developers because its syntax is similar to those languages. It offers fast performance and is designed to work well with web servers. It is widely used in building basic database-driven web applications.

PostgreSQL is another web database server that offers more advanced features often found in commercial database systems, such as transactions, subselects, triggers, views, referential integrity, and sophisticated locking. It is often used to provide more complex database functionality for websites and web applications.

Application Servers

In more sophisticated web application models, an application server is added to enable the system to manage business logic and track the user's progress through the application. The application server software runs in a middle tier, between web browser-based clients and back-end databases and business applications. The application server handles all of the application logic and connectivity that old-style client/server applications contained.

Examples of J2EE application servers is the open source JBoss application server.

1.1.5 General Web Services and Applications Architecture

The following diagram shows the basic architecture of the web components and services that are commonly used to host websites and build web applications.

Figure 1-1 *Architecture of Key Web Components and Technologies*
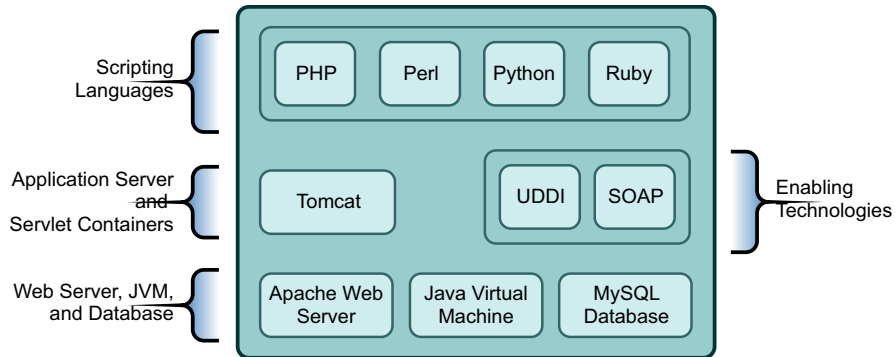
1.2 OES Components That Provide Web Services and Applications

OES comes bundled with all of the Web Services and Applications components you need to host dynamic web content and deploy web applications that you can either build yourself or download from the World Wide Web. Some of these components are developed by the open source software community, while others are developed by Novell. Each component offers an important building block that lets you build the solutions that best meet your business needs.

The following diagram illustrates how you can combine open source software and Novell software to provide web-based business solutions for employees, customers, and partners.

Figure 1-2   *Open Source and Custom Built Solutions*
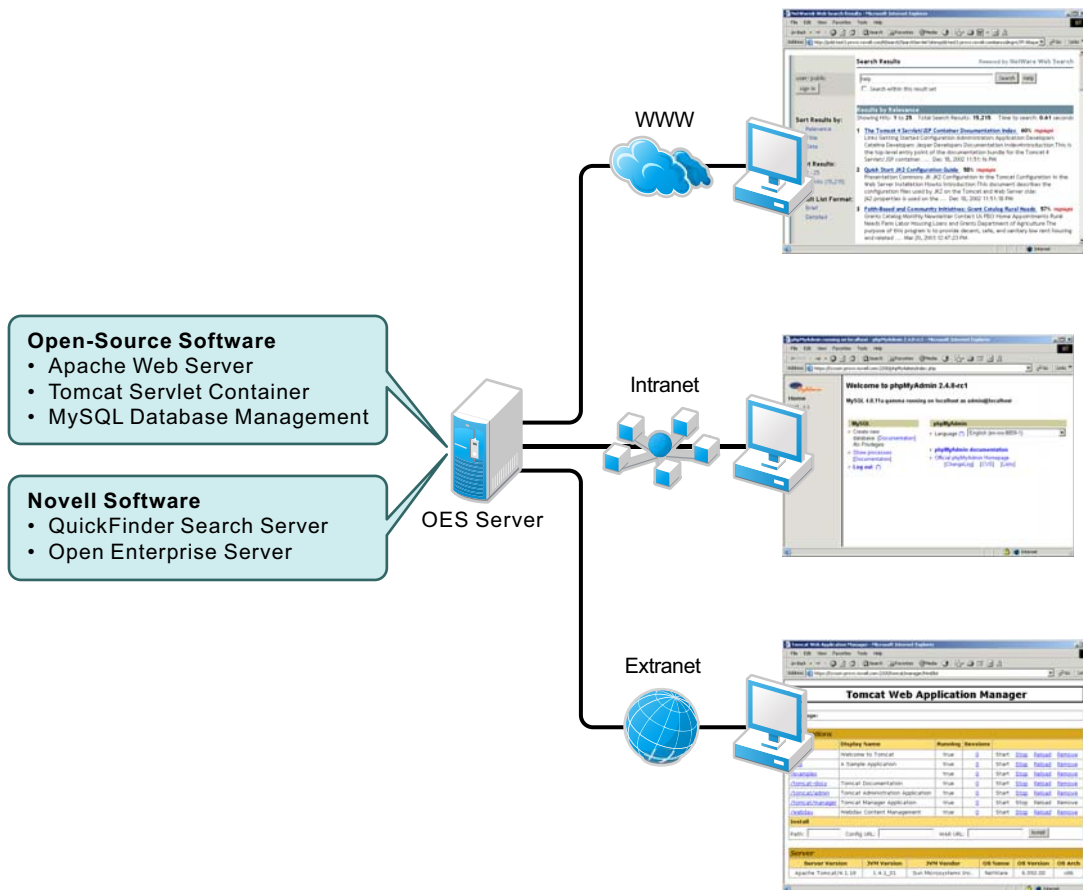
With the web components available in OES, you can:

* Host multiple websites on a single OES server.
* Manage all instances of the Apache web server from one interface using Apache Manager (regardless of what platform they are running on in your network).
* Choose from hundreds of free web applications that can be downloaded from the Internet and run on your OES server.
* Build and host your own web database applications.
* Choose from popular scripting languages to build your own dynamic web content.

- Build powerful web applications and services using the JBoss application server, which includes SOAP and UDDI components, as well as rapid application development support and application deployment capabilities.
- Add search and print functionality to any website, anywhere on the World Wide web or on a company intranet.

Some of the key benefits OES has to offer in the area of web and applications services include the following:

- Open source components that help you steer away from vendor lock-in and proprietary solutions. Applications that you develop can run on any other J2EE compliant platform, including UNIX and Windows operating systems.
- Valuable services for end users that enhance personal and team productivity.
- A strong J2EE and open source development model.
- A broad range of industry standard API sets.
- A broad selection of development tools and deployment models for developers. This provides tremendous flexibility in those cases where IT organizations decide to repurpose their servers.
- Lower IT spending because open source products are free and platform independent.

The following sections introduce each Web Services and Applications component included with OES:

1.2.1 Web Hosting: Apache HTTP Server

Apache is the most popular web server being used on the World Wide Web today. Its popularity comes from the fact that it is the most reliable and secure web server available. It runs on all major platforms, is capable of hosting even the most complex websites, and can scale to handle thousands of simultaneous connections.

The Apache Web Server serves as the foundation web server upon which you can build websites and host web applications for use in your business.

Key uses and benefits of using Apache in OES include the following:

- It provides a highly reliable and fast web server for hosting simple or complex websites.
- It is preconfigured to work with Jakarta-Tomcat, the servlet container created by the Apache Foundation, which can be used to host servlets and JavaServer Pages (JSPs) for automating business processes.
- It is ideal for web application development and testing.
- It lets you set up multiple virtual hosts for hosting multiple websites (with their own domain names) all from a single installation of Apache.

Figure 1-3 *Apache Running on an OES Server and Hosting Multiple website*
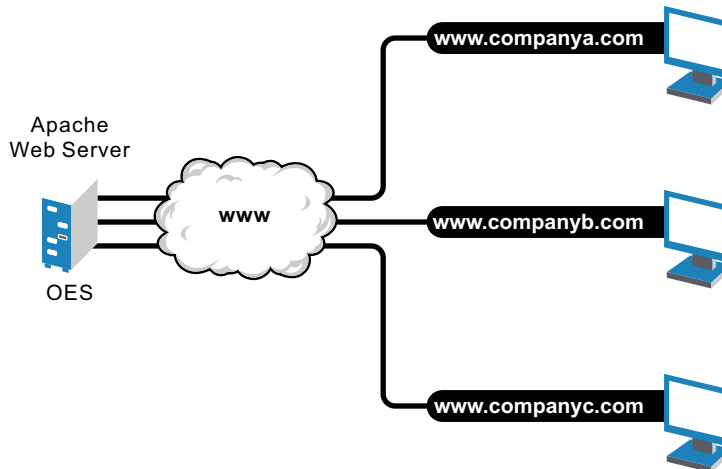
OES includes Apache Web Server 2.2 for Linux. It features a hybrid multi-process/multi-threaded implementation, filtering, simplified configuration, and a new API, along with extension modules to support Secure Sockets Layer (SSL), LDAP authentication, and multi-language error messages.

For information about using Apache HTTP Server on OES Servers and clustering website content with Novell Cluster Services, see Chapter 4, "Configuring Apache HTTP Server on OES Servers and in Clusters with Novell Cluster Services," on page 39.

1.2.2 Servlet Support: Tomcat Servlet Container

OES includes a Jakarta-Tomcat container for Linux. Tomcat is ideal for running basic Java servlet and JSP applications. OES also includes Tomcat 6 for Linux, which implements the Java Servlet and JSP specifications.

If you are relatively new to, or inexperienced with, Java programming and do not plan to build more advanced J2EE applications, the Tomcat container should satisfy your needs. It is very stable and includes all of the features of a commercial web application container.

1.2.3 Scripting: PHP and Perl

Scripting languages and visual builder tools have gained popularity in recent years because of their ease of use in delivering content to the web. OES provides a choice of scripting languages and the engines to run them. You can use these tools to develop web applications and administration utilities.

The scripting technologies integrated with OES Linux include industry standard PHP and Perl.

1.2.4 Web Databases: MySQL

OES includes the open source MySQL 5.0.67 database server on the Linux platform. When combined with a web application and a web server, MySQL is a very reliable and scalable database for use in hosting eCommerce and business-to-business web applications.

To manage your MySQL database, you can use the open source phpMyAdmin application written in the PHP language that provides a web-based administration tool.

The following diagram shows how MySQL can be used to host web database applications such as eCommerce or inventory tracking.

Figure 1-4 *MySQL and phpMyAdmin: Hosting Several Web Database Applications*
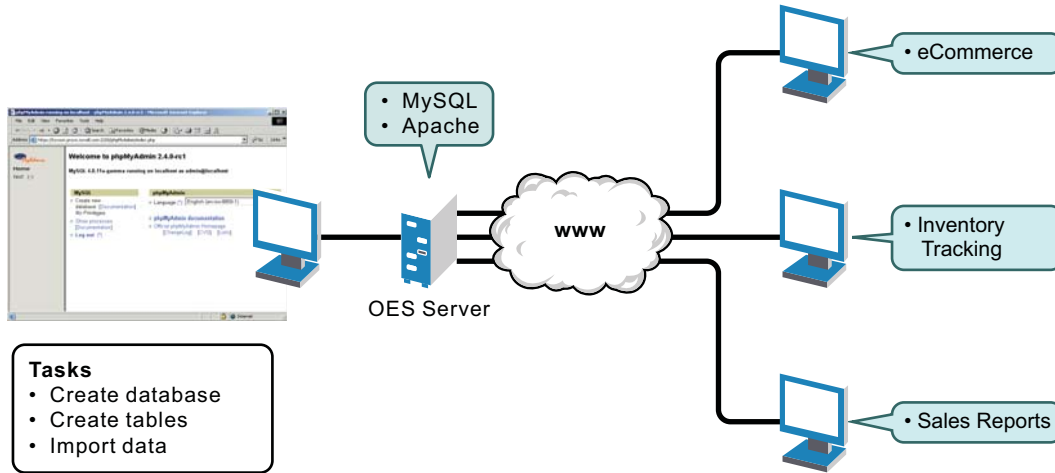
NOTE: PostgreSQL 9.1 database server comes with SUSE Linux Enterprise Server 11 SP3 and later.

1.2.5 Web Databases: PostgreSQL

SUSE Linux Enterprise Server 11 SP3 and later includes the open source PostgreSQL 8.3.*x* and 9.1 database servers. SUSE Linux Enterprise Server 11 SP2 and earlier includes the open source PostgreSQL 8.3.*x* database server. PostgreSQL is an advanced object-relational database management system that supports an extended subset of the SQL standard, including transactions, foreign keys, subqueries, triggers, and user-defined types and functions.

One PostgreSQL instance manages the data of one database. More then one PostgreSQL instance can run on a server at a time, but each must use a different data area and communication port.

Table 1-1 describes the two installation packages.

Table 1-1 *PostgreSQL Packages*

Package	Description
postgresql-server	This package includes the programs needed to create and run a PostgreSQL server. It allows you to create and maintain PostgreSQL databases.
postgresql	This package contains the basic utility and client programs necessary to maintain and work with local or remote PostgreSQL databases and the postgres(1), initdb(1), pg_ctl(1) manual pages for the SQL commands that PostgreSQL supports.

After you install the PostgreSQL software, you can use the PostgreSQL command-line commands, as described in Table 1-2, to create and manage the databases. See the `postgres(1)`, `initdb(1)`, `pg_ctl(1)` manual pages for information.

Table 1-2   PostgreSQL Commands

Command	Description
`postgres`	The `postgres` command is used to manage the PostgreSQL database server instance.
`initdb`	The `initdb` command is used to create a new PostgreSQL database.
`pg_ctl`	The `pg_ctl` command is used to stop, start, or restart a PostgreSQL server.

Full HTML documentation for PostgreSQL can be found in the `postgresql-docs` package. The start page is `file:///usr/share/doc/packages/postgresql/html/index.html`. The documentation is also available online at *PostgreSQL 8.3.x Documentation* (http://www.postgresql.org/docs/8.3/interactive/index.html) and *PostgreSQL 9.1 Documentation* (http://www.postgresql.org/docs/9.1/interactive/index.html)

IMPORTANT: The PostgreSQL server daemon uses SIGTERM to tell subordinate server processes to quit normally, and SIGQUIT to terminate without the normal cleanup. These signals should not be used by users. You should also avoid sending SIGKILL to the PostgreSQL server process, because it will interpret this as a crash and forces all the sibling processes to quit as part of its standard crash-recovery procedure.

1.2.6   Custom Web/J2EE Applications: WebSphere Application Server CE

When you need greater processing power beyond what scripting or web application hosting with Tomcat can offer, OES offers a J2EE-certified application servers: IBM WebSphere Application Server Community Edition Bundled with SLES 11, WebServer provides a J2EE-certified platform for building and deploying enterprise-class web applications.

IMPORTANT: The WebSphere Application Server CE package has been removed effective in SUSE Linux Enterprise Server 11 SP4.

IBM WebSphere Application Server Community Edition is an open source J2EE application server. Built on Apache Geronimo technology, it uses open source applications, such as Eclipse, Apache Tomcat, and Apache Derby, to provide an integrated foundation for developing and deploying Java applications. The embedded Apache Derby database provides a robust, small-footprint database server with full transactional capability. These components are described in Table 1-3.

Table 1-3   IBM WebSphere Application Server Components

Component	Description
Apache Geronimo v2.1.5	An open-source, JEE5 application server project that provides the foundation of the WebSphere Application Server Community Edition.
Apache Derby	A robust, small-footprint database server that is simple to deploy and reduces the cost of embedded and web-based applications.

Component	Description
Apache OpenEJB v3.0.2	An embeddable and lightweight EJB 3.0 implementation that can be used as a standalone server or embedded into Tomcat, JUnit, TestNG, Eclipse, IntelliJ, Maven, Ant, and any IDE or application.
Apache Tomcat v6.0.26	The servlet container that is used in the Reference Implementation for Java Servlet and JavaServer Pages technologies.
Eclipse plug-in	A plug-in used to develop, deploy, and debug J2EE applications to a Community Edition server.

Web-based tools based on the Eclipse platform provides a simple development environment for creating, deploying and debugging your WebSphere Application Server Community Edition applications.

For documentation, see the IBM WebSphere Application Server CE v2.1.1 User Guide (http://publib.boulder.ibm.com/wasce/Front_en.html).

1.3 What's Next

- To learn more about developing web applications for the OES environment, see the Novell Developer website.
- For general OES installation instructions for Linux, see the *OES 2015 SP1: Installation Guide*.
- For information about clustering a MySQL database with Novell Cluster Services, see Chapter 3, "Configuring MySQL with Novell Cluster Services," on page 21.
- For information about clustering web content with Novell Cluster Services, see Chapter 4, "Configuring Apache HTTP Server on OES Servers and in Clusters with Novell Cluster Services," on page 39.

2 What's New or Changed in OES Web Services

2.1 What's New or Changed in OES Web Services (OES 2015 SP1)

The web services and applications in Novell Open Enterprise Server (OES) 2015 SP1 comprise Novell software and open source software that support SUSE Linux Enterprise Server (SLES) 11 Service Pack 4 (SP4). There are no new features or enhancements in OES 2015 SP1.

2.2 What's New or Changed in OES Web Services (OES 2015)

The web services and applications in Novell Open Enterprise Server (OES) 2015 comprise Novell software and open source software that support SUSE Linux Enterprise Server (SLES) 11 Service Pack 3 (SP3). There are no new features or enhancements in OES 2015.

3 Configuring MySQL with Novell Cluster Services

Open Enterprise Server 2015 SP1 provides an open source version of the MySQL 5.5.*x* software that is offered under the GNU General Public License (GPL) Version 2. Version 5.0.*x* is available on SLES 11 SP2 and earlier. MySQL can be used with Novell Cluster Services to provide high availability support to the customers you service with MySQL. This helps prevent interruptions of access for the MySQL database.

IMPORTANT: As stated in the *Release Notes for SUSE Linux Enterprise Server 11 and later* (http:// www.novell.com/linux/releasenotes/x86_64/SUSE-SLES/11-SP3/), the open source MySQL packages require additional support contracts to be obtained by the customer in order to receive full support.

The MySQL database format is upgraded from version 5.0 to version 5.5 in SUSE Linux Enterprise Server 11 SP4 and OES 2015 SP1. For information about upgrading the database format, see "Upgrading from MySQL 5.0 to MySQL 5.5 Introduces a New Database Format" on page 57.

MySQL is installed on all nodes where you want it to run, but a database runs on only one node in the cluster at a time. The MySQL configuration files are modified on each node to point to a path on a Linux Logical Volume Manager (LVM) volume group cluster resource that contains the MySQL database files. You cluster-enable the volume group by using the MySQL template, then configure its resource load, unload, and monitoring scripts, set its resource failover and failback modes, and assign the resource to specific nodes in the cluster. When a node fails where the resource is online, the resource fails over to the next preferred node in the cluster.

IMPORTANT: Refer to the official MySQL 5.5 documentation for information about configuring, managing, and using MySQL. For information, see the MySQL Documentation Library: MySQL Reference Manuals (http://dev.mysql.com/doc/).

The instructions in this section describes how to set up MySQL in a Novell Cluster Services cluster.

3.1 Prerequisites for Clustering MySQL

The following setup is required for clustering the MySQL database files with Novell Cluster Services:

❏ Novell Cluster Services must be installed and configured as described in "Installing, Configuring, and Repairing Novell Cluster Services" in the *OES 2015 SP1: Novell Cluster Services for Linux Administration Guide*.

❏ MySQL must be installed on every node in the cluster where you want MySQL to run. The installation is described in Section 3.2, "Installing and Enabling MySQL," on page 22.

❏ The SAN device that you want to use for the MySQL database must be accessible to all nodes in the cluster. It will be activated on only one node at a time.

❏ You must create a shared Linux Logical Volume Management (LVM) volume group where you will store the MySQL database and configuration file. This setup is described in Section 3.3, "Creating an LVM Volume Group and Logical Volume," on page 24.

3.2 Installing and Enabling MySQL

Before you configure MySQL with Novell Cluster Services, MySQL must be installed and configured properly on all servers in the cluster where you intend to run it. You can use the YaST Software Management tool to install the MySQL and the MySQL Client packages. Other MysQL packages are available that allow you to use MySQL with Perl, PHP, Postfix, or Python, but this guide does not cover their installation or use.

Package	Description
`mysql`	Provides the MySQL software and database.
`mysql-client`	Provides the MySQL client command line program that acts as a text-based front end for the MySQL Server. It's used for issuing queries and viewing the results interactively from a terminal window
`mysql-Max`	Provides the MySQL software, database, and the following features for users that require transaction support: ◆ Berkeley database (BDB) tables ◆ InnoDB tables These features provide transaction-safe tables to which locks are applied while a series of SQL queries is made. The series of queries is referred to as a transaction.
`perl-DBD-mysql`	Provides a MySQL database driver (DBD) to support a database-independent interface (DBI) for the Perl programming language.
`php5-mysql`	Provides a PHP plug-in that allows an Apache HTTP server to access a MySQL database.
`postfix-mysql`	Provides a Postfix plug-in that allows a Postfix mail system to access a MySQL database.
`python-mysql`	Provides a Python plug-in that allows you to execute SQL queries on a MySQL database through your Python application.

Use the following procedure to install the `mysql` and `mysql-client` packages, and enable MySQL on each node in the cluster:

1 Log in to the server as the Linux `root` user, then open YaST.

2 Ensure that the SUSE Linux Enterprise Server 11 SP*x* installation CD is mounted on the server.

3 In YaST, select **Software > Software Management**, then click the **Search** tab.

4 To find the components, type `mysql` in the **Search** field, then click **Search**.

5 In the **Package** list, select `mysql` and `mysql-client`.
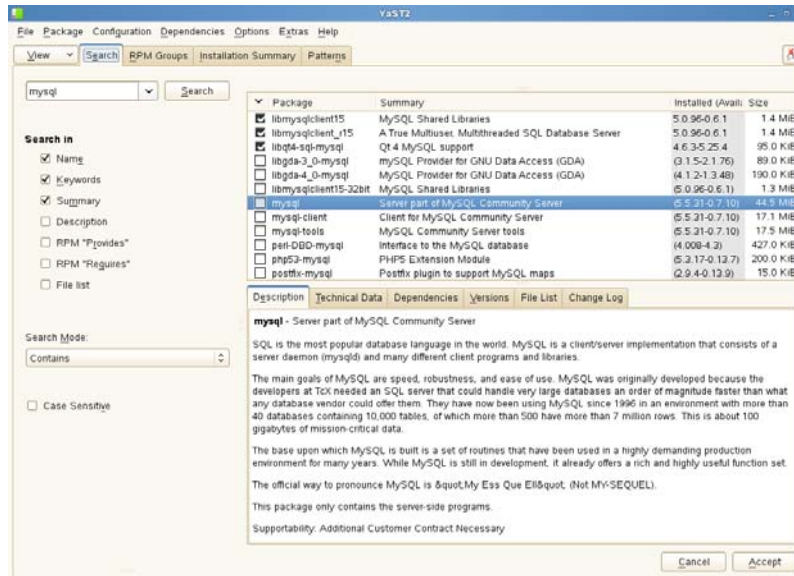
6 Click **Accept**, then click **Continue** for each component to confirm that you want to install it.

YaST does the following:

- Installs the MySQL Server and MySQL Client software.

   The software is not enabled by default, and the MySQL daemon is not running at this time. No run levels are set.

- Creates the MySQL root user (a user internal to the MySQL system) as a superuser that has access rights to perform any function in MySQL. Initially, this user has no password assigned.

- Creates a default path `/var/lib/mysql` for storing databases that you create later. Initially, this directory is empty. It is populated later when you enable the MySQL service.

- Creates the `mysql` user and group on the server and makes them the owners of the default data directory `/var/lib/mysql` and its contents.

- Creates a default mount point `/mnt/mysql` for the database. This is where you will mount the LVM logical volume that you create for the database in Section 3.3, "Creating an LVM Volume Group and Logical Volume," on page 24.

- Creates the default MySQL configuration file (`/etc/my.cnf`).

7 In YaST, enable the MySQL service:

7a Select **System > System Services (Runlevel)**.

7b Select **Expert Mode**.

7c In the **Service** list, select the `mysql` daemon.

7d Click **Set/Reset > Enable the Service**.

Under **Service will be started in the following runlevels**, notice that the 2, 3, and 5 check boxes are selected by default. You don't want the service to start on system boot because it starts when the cluster resource is brought online on a cluster node.
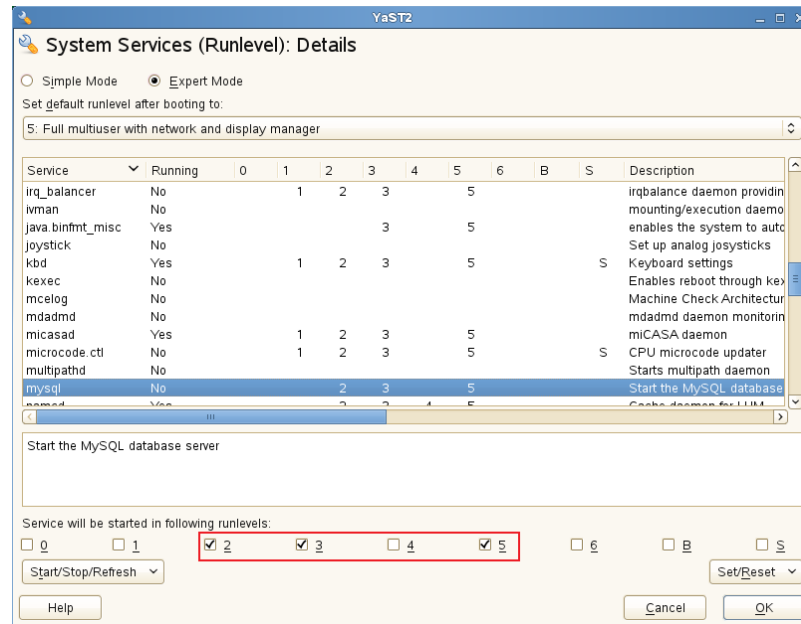
7e In the lower right corner, click **OK**.

7f When you are prompted to confirm the changes, click **Yes** to save them.

7g Exit YaST.

8 Stop the MySQL daemon from running. In a terminal console, enter the following as the Linux `root` user:

```
/etc/init.d/mysql stop
```

Alternatively, you can use the `rcmysql stop` command.

9 Repeat Step 1 through Step 8 on each node in the cluster to install and enable MySQL.

10 After you have installed and enabled MySQL on all servers in the cluster, continue with Section 3.3, "Creating an LVM Volume Group and Logical Volume," on page 24.

3.3 Creating an LVM Volume Group and Logical Volume

After you have installed MySQL, you are ready to set up the LVM volume group and logical volume where you will store a MySQL database. Sample values are used in the procedures in this section to help you understand what is required at each step. The overview provides only the Linux commands that you need to create and prepare the volume group for use by MySQL. The detailed description provides more information about the process, including the syntax and sample commands.

- Section 3.3.1, "Sample Values," on page 25
- Section 3.3.2, "Setting Up the VG and LV (Overview)," on page 25
- Section 3.3.3, "Setting up the VG and LV (Detailed)," on page 26

3.3.1   Sample Values

The procedures in this section uses the following parameters. Ensure that you replace the sample values with your values. The first node in the cluster is where you configure MySQL and the cluster resource.

Parameter	Sample Value
LVM physical volume	`/dev/sdd`
LVM volume group name	`mysqlvg`
LVM logical volume	`msqllv`
File system type	`ext3` This is the file system type that you make on the LVM logical volume, such as `ext2`, `ext3`, `reiserfs`, or `xfs`.
Logical volume path	`/dev/mysqlvg/mysqllv`
Mount point for the logical volume	`/mnt/mysql`
Default MySQL root path	`/var/lib/mysql`
New MySQL root path	`/mnt/mysql/var/lib/mysql`

3.3.2   Setting Up the VG and LV (Overview)

You can create the volume group and logical volume by issuing the following LVM commands as the *root* user on the cluster node. This overview of the process uses the sample values. Ensure that you substitute your own values in the commands. For details, see Section 3.3.3, "Setting up the VG and LV (Detailed)," on page 26.

Command Action	Command
1. Create the LVM physical volume.	`pvcreate /dev/sdd`
2. Create the clustered LVM volume group.	`vgcreate -c y mysqlvg /dev/sdd`
3. Activate the volume group exclusively on the node.	`vgchange -a ey mysqlvg`
4. Create the LVM logical volume.	`lvcreate -n mysqllv -L size mysqlvg`
5. Add a file system to the LVM logical volume.	`mkfs -t ext3 /dev/mysqlvg/mysqllv [fs_options]`
6. Create a mount point for the logical volume.	`mkdir /mnt/mysql` You must also create this path on each node in the cluster.
7. Mount the LVM logical volume.	`mount -t ext3 /dev/mysqlvg/mysqllv /mnt/mysql`

8. Create the directory structure for the MySQL database files on the mounted logical volume.	`cd /mnt/mysql`
	`mkdir /mnt/mysql/var`
	`mkdir /mnt/mysql/var/lib`
	`mkdir /mnt/mysql/var/lib/mysql`

9. Modify the file ownership of the mount point and subdirectories.	`chown -R mysql:mysql /mnt/mysql`

Command Action	Command
10. Unmount the LVM logical volume.	`cd /` `unmount /mnt/mysql`
10. Deactivate the LVM logical volume.	`vgchange -a n mysqlvg`

3.3.3 Setting up the VG and LV (Detailed)

For detailed instructions, use the following procedure to create the LVM volume group and logical volume:

1 Log in as the Linux `root` user to the first node of the cluster, then open a terminal console.

2 In NSSMU, initialize the SAN device that you want to use for the MySQL database, but do not mark it as shareable for clustering:

  2a At the console prompt, launch NSSMU by entering:

    `nssmu`

  2b Select **Devices**, then press Enter.

  2c In the **Devices** list, select the unpartitioned device that you want to use, then press F3 to initialize it.

  2d Read the advisory, then press Y to confirm that you want to initialize the device.

  2e Specify the Master Boot Record (MBR) type as DOS or GPT, then press Enter.

    Typically, you use DOS format for devices up to 2 TB. You use GPT for devices greater than 2 TB.

  2f Verify that the device is initialized and that it is unshared (that is, **Shareable for Clustering** is set to **No**).
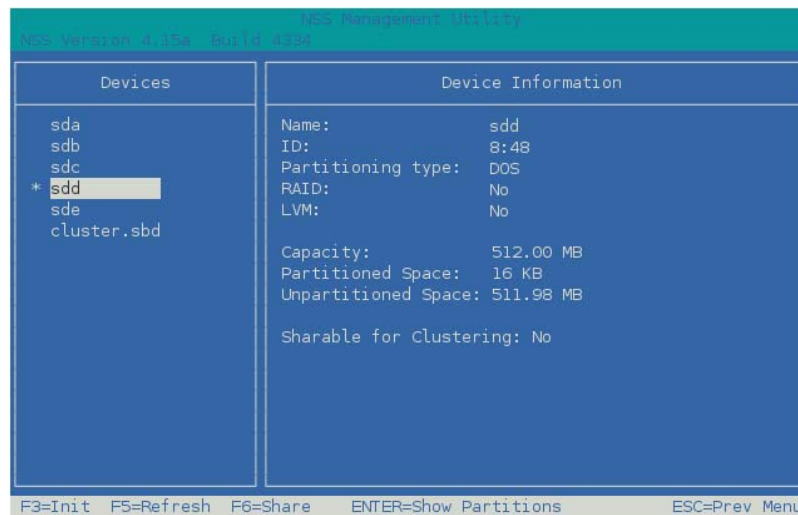
  2g Exit NSSMU to return to the command prompt.

3 Create an LVM physical volume on the device (such as `/dev/sdd`) by entering:

  `pvcreate <device>`

For example:

```
pvcreate /dev/sdd
  No physical volume label read from /dev/sdd
  Physical volume "/dev/sdd" successfully created
```

4 Create an LVM volume group (such as `mysqlvg`) on the physical volume by entering:

```
vgcreate -c y <vg_name> <device>
```

For example:

```
vgcreate -c y "mysqlvg" /dev/sdd
  Clustered volume group "mysqlvg" successfully created
```

The volume group is automatically activated.

5 Activate the volume group exclusively on the current server by entering:

```
vgchange -a ey <vg_name>
```

The -a option activates the volume. The `ey` parameter specifies the values `exclusively` and `yes`.

For example:

```
vgchange -a ey mysqlvg
```

6 View information about the volume group by using the `vgdisplay` command:

```
vgdisplay <vg_name>
```

Notice that 4 MB of the device are used for the volume group's Physical Extent (PE) table. You must consider this reduction in available space on the volume group when you specify the size of the LVM logical volume in the next step (Step 7).

For example:

```
vgdisplay mysqlvg
  --- Volume group ---
  VG Name               mysqlvg
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               508.00 MB
  PE Size               4.00 MB
  Total PE              127
  Alloc PE / Size       0 / 0
  Free  PE / Size       127 / 508.00 MB
  VG UUID               rqyAd3-U2dg-HYLw-0SyN-loO7-jBH3-qHvySe
```

7 Create an LVM logical volume (such as `mysqllv`) on the volume group by entering:

```
lvcreate -n <lv_name> -L size <vg_name>
```

Specify the logical volume name, size, and the name of the volume group where you want to create it. The size is specified in megabytes by default.

The logical volume full path name is /dev/*<vg_name>*/*<lv_name>*.

For example:

```
lvcreate -n "mysqllv" -L 500 "mysqlvg"
  Logical volume "mysqllv" created
```

This volume's full path name is /dev/mysqlvg/mysqllv.

8 View information about the logical volume by entering:

```
lvdisplay -v <lv_path_name>
```

For example:

```
lvdisplay -v /dev/mysqlvg/mysqllv
    Using logical volume(s) on command line
  --- Logical volume ---
  LV Name               /dev/mysqlvg/mysqllv
  VG Name               mysqlvg
  LV UUID               nIfsMp-alRR-i4Lw-Wwdt-v5io-2hDN-qrWTLH
  LV Write Access       read/write
  LV Status             available
  # open                0
  LV Size               500.00 MB
  Current LE            125
  Segments              1
  Allocation            inherit
  Read ahead sectors    auto
  - currently set to    1024
  Block device          253:1
```

9 Create a file system (such as Ext2, Ext3, ReiserFS, or XFS) on the LVM logical volume by entering:

```
mkfs -t <fs_type> <lv_path_name> [fs_options]
```

You can specify file system options according to the type of file system you are making. For information, see the mkfs(8) man page and the related man page for the file system type, such as mkfs.ext2(8), mkfs.ext3(8), mkfs.reiserfs(8), or mkfs.xfs(8).

For example:

```
mkfs -t ext3 /dev/mysqlvg/mysqllv
  mke2fs 1.41.9 (22-Aug-2009)
  Filesystem label=
  OS type: Linux
  Block size=1024 (log=0)
  Fragment size=1024 (log=0)
  128016 inodes, 512000 blocks
  25600 blocks (5.00%) reserved for the super user
  First data block=1
  Maximum filesystem blocks=67633152
  63 block groups
  8192 blocks per group, 8192 fragments per group
  2032 inodes per group
  Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409
  Writing inode tables: done
  Creating journal (8192 blocks): done
  Writing superblocks and filesystem accounting information: done

  This filesystem will be automatically checked every 29 mounts or
    180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

10 Create a mount point for the logical volume by entering:

```
mkdir /mnt/mysql
```

11 Mount the logical volume on the MySQL mount point by entering:

```
mount -t <fs_type> <lv_path_name> <mount_point>
```

For example:

```
mount -t ext3 /dev/mysqlvg/mysqllv /mnt/mysql
```

12 Go to the mount point location (`/mnt/mysql`), then create the `/var/lib/mysql` subdirectory structure by entering:

```
cd /mnt/mysql
mkdir /mnt/mysql/var
mkdir /mnt/mysql/var/lib
mkdir /mnt/mysql/var/lib/mysql
```

13 Change the owner and group owner of the `/mnt/mysql` directory and its contents to use the `mysql` user and group. Enter the `chown` command with the recursive (-R) option:

```
cd /mnt
chown -R mysql:mysql mysql
```

Another way to do this is to explicitly specify the directory path:

```
chown -R mysql:mysql /mnt/mysql
```

14 Continue with Section 3.4, "Configuring MySQL on the LVM Logical Volume," on page 30.

3.4 Configuring MySQL on the LVM Logical Volume

MySQL databases are usually located in a subdirectory of the `/var/lib/mysql/` directory. If you create a database named `test`, then the database files are located in the `/var/lib/mysql/test` directory.

In order for MySQL to take advantage of the benefits provided by Novell Cluster Services, you must make some configuration changes to MySQL. On the first server, you copy the default MySQL configuration file (`/etc/my.cnf`) to the LVM logical volume, modify `/mnt/mysql/var/lib/mysql/my.cnf` file so that all `datadir` entries are commented out, then create a MySQL database on the LVM Logical volume.

The following instructions assume that you have not created a database on the server at this time. If a MySQL database currently exists in the default `/var/lib/mysql` location, the database's directory and its contents must be relocated to the new `/mnt/mysql/var/lib/mysql` path, rather than creating it as described in of the following procedure. Afterwards, Ensure that you modify the ownership of the folder and files to the `mysql` user and group by using the `chown` command as illustrated in of Section 3.3, "Creating an LVM Volume Group and Logical Volume," on .

IMPORTANT: After you have modified the MySQL configuration file to use the LVM logical volume path, you should always exclusively activate the volume group on the server before attempting to start the MySQL daemon. The cluster resource does this automatically in the load script.

To configure a MySQL database on the LVM logical volume:

1 Log in as the Linux `root` user on the first node, then open a file browser or terminal console.

2 Copy the default `/etc/my.cnf` configuration file to the `/mnt/mysql/var/lib/mysql` directory. Enter

```
cp /etc/my.cnf /mnt/mysql/var/lib/mysql
```

3 In a text editor, modify the `/mnt/mysql/var/lib/mysql/my.cnf` file and comment out any data directory entries, then save your changes.

```
# datadir=
```

4 Change the permissions on the `/mnt/mysql/var/lib/mysql/my.cnf` file to Read and Execute for each permission level, and change the ownership to the `mysql` user and group. Enter the following commands:

```
chmod 555 /mnt/mysql/var/lib/mysql/my.cnf
chown mysql:mysql /mnt/mysql/var/lib/mysql/my.cnf
```

You can view these settings by using the `ll` `<filepath>` command. For example:

```
ll /mnt/mysql/var/lib/mysql/my.cnf
  -r-xr-xr-x 1 mysql mysql 6297 2011-07-08 14:19 /mnt/mysql/var/lib/mysql/
my.cnf
```

5 Open a terminal console as the Linux `root` user, then start MySQL:

```
/etc/init.d/mysql start
```

Another option is to use the `rcmysql start` command.

6 Create a database named `data` on the LVM logical volume:

```
mysql_install_db --datadir=/mnt/mysql/var/lib/mysql/data --user=mysql
```

7 Set the ownership of the `data` database to be the `mysql` user and group:

```
chown -R mysql:mysql /mnt/mysql/var/lib/mysql/data
```

8 Stop the MySQL daemon from running:

```
/etc/init.d/mysql stop
```

Another option is to use the `rcmysql stop` command.

9 Deactivate the LVM volume group:

```
vgchange -a n <vg_name>
```

For example:

```
vgchange -a n mysqlvg
```

10 Continue with .

3.5 Cluster-Enabling MySQL on the Logical Volume

Now that you have configured MySQL for the LVM logical volume, you are ready to cluster-enable MySQL. In iManager, you use the Novell Cluster Services MySQL template to create a cluster resource for the LVM volume group that contains the MySQL database. The resource's load script starts the MySQL daemon when the resource is brought online, and the unload script stops it when the resource is taken offline.

The sample scripts in this section use the following sample parameters. Ensure that you replace the sample values with your values.

Parameter	Sample Value
Resource IP Address	10.10.10.44
MOUNT_FS	ext3
	This is the file system you created on the LVM volume group, such as ext2, ext3, reiserfs, or xfs.
VOLGROUP_NAME	`mysqlvg`
MOUNT_DEV	`/dev/$VOLGROUP_NAME/mysqllv`
MOUNT_POINT	`/mnt/mysql`
MySQL_ROOT	`$MOUNT_POINT/var/lib/mysql`

Use the following procedure to create the MySQL cluster resource for the LVM volume group:

1 In iManager, select **Clusters > My Cluster**, select the cluster., then select the **Cluster Options** tab.

2 Under the **Cluster Objects** title, click **New**.

3 On the **New Resource > Resource Type** page, specify **Resource** as the type, then click **Next**.

4 On the **New Resource > Cluster Resource Information** page, specify a cluster resource name, browse to select the **MySQL_Template,** then click **Next**.

Do not select **Online Resource after Create**. You must configure the resource scripts and settings before bringing the resource online.

5 On the Load Script page, modify the definition fields for your MySQL resource, file system type, volume group name, logical volume name, and mount point, then click **Next**.

The following load script uses the sample values from the MySQL setup:

```
#!/bin/bash
. /opt/novell/ncs/lib/ncsfuncs

# define the IP address
RESOURCE_IP=10.10.10.44
# define the file system type
MOUNT_FS=ext3
#define the volume group name
VOLGROUP_NAME=mysqlvg
# define the device
MOUNT_DEV=/dev/$VOLGROUP_NAME/mysqllv
# define the mount point
MOUNT_POINT=/mnt/mysql

# define MySQL database root
MySQL_ROOT=$MOUNT_POINT/var/lib/mysql

#activate the volume group
exit_on_error vgchange -a ey $VOLGROUP_NAME

# mount the file system
exit_on_error mount_fs $MOUNT_DEV $MOUNT_POINT $MOUNT_FS

# add the IP address
exit_on_error add_secondary_ipaddress $RESOURCE_IP

# start MySQL
/usr/bin/mysqld_safe --user=mysql --pid-file=$MySQL_ROOT/mysql.pid --
socket=$MySQL_ROOT/mysql.sock --datadir=$MySQL_ROOT --bind-
address=$RESOURCE_IP &>/dev/null &

# return status
exit 0
```

6 On the Unload Script page, modify the definition fields for your MySQL resource, file system type, volume group name, logical volume name, and mount point, then click **Next**.

The following unload script uses the sample values from the MySQL setup:

```
#!/bin/bash
. /opt/novell/ncs/lib/ncsfuncs

# define the IP address
RESOURCE_IP=10.10.10.44
# define the file system type
MOUNT_FS=ext3
#define the volume group name
VOLGROUP_NAME=mysqlvg
# define the device
MOUNT_DEV=/dev/$VOLGROUP_NAME/mysqllv
# define the mount point
MOUNT_POINT=/mnt/mysql

# define MySQL database root
MySQL_ROOT=$MOUNT_POINT/var/lib/mysql

# request MySQL stop
ignore_error killproc -p $MySQL_ROOT/mysql.pid -TERM /usr/sbin/mysqld

# del the IP address
ignore_error del_secondary_ipaddress $RESOURCE_IP

# umount the file system
sleep 10 # if not using SMS for backup, please comment out this line
exit_on_error umount_fs $MOUNT_DEV $MOUNT_POINT $MOUNT_FS

#deactivate the volume group
exit_on_error vgchange -a n $VOLGROUP_NAME

# return status
exit 0
```

7 On the Monitoring Script page, modify the definition fields for your MySQL resource, file system type, volume group name, logical volume name, and mount point, then click **Next**.

The following monitoring script uses the sample values from the MySQL setup:

```
#!/bin/bash
. /opt/novell/ncs/lib/ncsfuncs

# define the IP address
RESOURCE_IP=10.10.10.44
# define the file system type
MOUNT_FS=ext3
#define the volume group name
VOLGROUP_NAME=mysqlvg
# define the device
MOUNT_DEV=/dev/$VOLGROUP_NAME/mysqllv
# define the mount point
MOUNT_POINT=/mnt/mysql

# define MySQL database root
MySQL_ROOT=$MOUNT_POINT/var/lib/mysql

#check the logical volume
```

```
exit_on_error status_lv $MOUNT_DEV

# check the file system
exit_on_error status_fs $MOUNT_DEV $MOUNT_POINT $MOUNT_FS

# check the IP address
exit_on_error add_secondary_ipaddress $RESOURCE_IP

# check MySQL
exit_on_error checkproc -p $MySQL_ROOT/mysql.pid /usr/sbin/mysqld


# return status
exit 0
```

8 On the Resource Policies page, specify the **Resource Behavior**, **Start Mode**, **Failover Mode**, and **Failback Mode**, then click **Next**.

For information about completing these fields, see "Configuring the Start, Failover, and Failback Modes for Cluster Resources" in the *OES 2015 SP1: Novell Cluster Services for Linux Administration Guide*.

9 On the Resource Preferred Nodes page, assign the nodes where MySQL is installed, then click **Finish**.

The resource appears in the Cluster Objects list:

10 Bring the MySQL resource online. Select **Cluster Manager**, select the MySQL resource check box, then click **Online**.

If the resource goes comatose, offline the resource, then open its properties page and re-verify the scripts.

3.6 File Location

During the MySQL installation, the following files are unpackd or created by YaST:

MySQL Component	Default Location in OES
MySQL daemon for start, stop, and restart commands	`/etc/init.d/mysql`
Configuration files	`/etc/my.cnf` `/etc/mysqlaccess.conf`
Database files	`/var/lib/mysql`
Man pages	`/usr/share/man/man1`
Documentation (*MySQL Readme*)	`/usr/share/doc/packages/mysql`
Log file	`/var/lib/mysql/mysqld.log` The MySQL log file can also be accessed via a hard link from `/var/log/mysqld.log`.

Software	Some of the software components might not appear in this location until after you enable the service.
	
	`/usr/bin/mysql`
	`/usr/bin/mysqladmin`
	`/usr/bin/mysqlbinlog`
	`/usr/bin/mysqlbug`
	`/usr/bin/mysqlcheck`
	`/usr/bin/mysqld_multi`
	`/usr/bin/mysqld_safe`
	`/usr/bin/mysqldump`
	`/usr/bin/mysqldumpslow`
	`/usr/bin/mysql_fix_extensions`
	`/usr/bin/mysql_fix/privilege_tables`
	`/usr/bin/mysqlimport`
	`/usr/bin/mysql_install_db`
	`/usr/bin/mysql_secure_installation`
	`/usr/bin/mysqlshow`
	`/usr/bin/mysqlupgrade`
	`/usr/bin/my_print_defaults`
	`/usr/bin/myisamcheck`
	`/usr/bin/myisam_ftdump`
	`/usr/bin/myisamlog`
	`/usr/bin/myisampack`

3.7 Security Considerations for the MySQL Configuration

Consider the security measures in this section when working with MySQL.

3.7.1 MySQL Ports

MySQL uses port 3306 by default. Additional ports are assigned sequentially as 3307, 3308, and so on. These ports must be open in the firewall in order to allow remote access to the MySQL database.

3.7.2 Securing MySQL

The default installation of MySQL provides some configuration settings, an anonymous user, and the `test` database that can possibly compromise security in a production environment:

- The root user can connect from the local host or remotely.
- An anonymous user is also created and can connect from the local host or remotely.
- Any local user on the server can connect to the `test` database without a password and be treated as the anonymous user.
- The anonymous user can perform any function on any databases named `test` or with a name that begins with `test_`.

For production servers, we recommend that you secure your MySQL service by setting a password for the MySQL root user. This is a password for the MySQL administrator user, that is, a root user within the MySQL system. It is not the Linux root user.

1 To set the password and log in to MySQL on the server, enter the following commands:

```
/usr/bin/mysqladmin -u root password <new_password>

/usr/bin/mysqladmin -u root -h <server_fdn_name> password <new_password>
```

For example:

```
/usr/bin/mysqladmin -u root password novell
/usr/bin/mysqladmin -u root -h myserver1.europe.example.com password novell
```

Alternatively, you can run the `mysql_secure_installation` command as the Linux `root` user, complete the fields that make sense for your MySQL configuration, then use:

```
/usr/bin/mysql_secure_installation
```

We recommend that you configure the following secure settings:

- Set a password for the MySQL root user.
- Remove MySQL anonymous users.
- Disallow remote login for the MySQL root user.

  The MySQL root user is allowed to connect to the database, but only from the local host.
- Remove the `test` database.
- Reload the Privileges table.

3.8 Additional Information

The following resources are available to help you manage and use MySQL:

- *MySQL 5.5 Reference Manual* (http://dev.mysql.com/doc/refman/5.5/en/index.html) from the MySQL Documentation Library (http://dev.mysql.com/doc/)
- After you have installed MySQL and the MySQL client on the server, the following man pages are available for MySQL utilities by entering the `man <mysql_utility>` command:
  - `mysql(1)`
  - `mysqladmin(1)`
  - `mysqlbinlog(1)`

- mysqlbug(1)
- mysqlcheck(1)
- mysqld_multi(1)
- mysqld_safe(1)
- mysqldump(1)
- mysql_fix_extensions(1)
- mysql_fix_privileg_tables(1)
- mysqlimport(1)
- mysql_install_db(1)
- mysqlsecure_installation(1)
- mysqlshow(1)
- mysqlupgrade(1)

4 Configuring Apache HTTP Server on OES Servers and in Clusters with Novell Cluster Services

The Apache HTTP Server is an open source web server developed by the Apache Software Foundation (http://www.apache.org). On an Open Enterprise Server (OES) 11 or later cluster, you can use Novell Cluster Services to cluster the web content for your personalized websites. The Apache service is not cluster aware and must run on each server in the cluster.

Clustering your website content helps make your website highly available for your customers. With Novell Cluster Services, if your web server fails, you can use any of the servers in the cluster to host your website, which results in virtually zero down-time for your customers.

This section describes key considerations for configuring the Apache virtual hosts for your personalized websites.

- Section 4.1, "Prerequisites for Using Apache on OES Servers," on page 39
- Section 4.2, "Prerequisites for Using Apache in NCS Clusters," on page 40
- Section 4.3, "Understanding the Default OES Setup of Apache HTTP Server," on page 41
- Section 4.4, "Using Apache HTTP Server on OES Servers (Single Server or Cluster Nodes)," on page 43
- Section 4.5, "Troubleshooting the Apache HTTP Server," on page 54
- Section 4.6, "Additional Information," on page 55

4.1 Prerequisites for Using Apache on OES Servers

The following setup is required to use Apache HTTP Server on your OES servers:

- When you install OES services on the server, Novell-ready versions of Apache 2 (Prefork, 64-bit) and Tomcat 6 are automatically installed and configured. You manually manage Apache services with the Apache configuration files. Use a text editor to create or modify the configuration files, then gracefully restart the Apache HTTP Server daemon (`rcapache2 graceful`) to apply the changes.

  WARNING: Do not install the default Linux **Web and LAMP** pattern independently of the OES patterns being added to the server. Do not use the **HTTP Server** option in YaST to configure Apache or virtual host settings on an OES server. It overwrites essential OES settings for Apache and breaks the existing setup. For recovery information, see Section 4.5.1, "Apache Server Errors after Using the HTTP Server Option in YaST," on page 54.

  OES selects the **Linux Web and LAMP** pattern and configures the HTTP server settings for OES when you install any of the OES services from the OES Add-On DVD. If you plan to install OES services after the SLES installation, do not install the Linux Web and LAMP pattern as part of the SLES setup, doing so will configure the HTTP server with the SLES defaults. Then, when you add OES services to the server, the HTTP server setup might be broken for OES. For possible

workarounds to fix the HTTP server configuration to work for OES, see the default OES setup described in Section 4.3, "Understanding the Default OES Setup of Apache HTTP Server," on page 41 and Section 4.5, "Troubleshooting the Apache HTTP Server," on page 54.

- You can use the Novell Storage Services (NSS) file system or Linux file systems to host your web content:

  - **NSS volumes:** Install the **Novell Storage Services** pattern on the server. For information, see "Installing and Configuring Novell Storage Services" in the *OES 2015 SP1: NSS File System Administration Guide for Linux*.

    Novell Cluster Services supports cluster resources for NSS pools. For information about creating clustered NSS pools and volumes, see "Configuring and Managing Cluster Resources for Shared NSS Pools and Volumes" in the *OES 2015 SP1: Novell Cluster Services for Linux Administration Guide*.

  - **Linux volumes:** The Linux Logical Volume Manager (LVM) and POSIX file systems (such as Btrfs, Ext2, Ext3, Reiser, and XFS) are installed automatically. You can manage the volumes with NSSMU and Novell Linux Volume Manager (NLVM) if you install Novell Storage Services on the server.

    To use NCP-enabled Linux volumes, install NCP Services and Novell Remote Manager on each OES node in the cluster. For information, see the *OES 2015 SP1: NCP Server for Linux Administration Guide*.

- You can host multiple websites on the same server. You must configure an Apache virtual host for each website.

- The following permissions are required:

  - The user `wwwrun` must be the file owner of the website directories and files. The group can be the system `root` or the Apache group `www`.

  - If web content resides on an NSS volume or on an NCP-enabled Linux volume, the following additional permissions are required:

    - Enable the eDirectory user `wwwrun` and group `www` with Linux User Management (LUM). OES automatically creates and LUM-enables the user and group when you install the first OES server in a NetIQ eDirectory tree.

    - Assign the eDirectory user `wwwrun` as a file system trustee with Read and File Scan rights for the directory you specify in the `DocumentRoot` directive in the virtual host configuration file.

For information about the default OES setup for Apache and setting up virtual hosts, see Section 4.4, "Using Apache HTTP Server on OES Servers (Single Server or Cluster Nodes)," on page 43.

4.2 Prerequisites for Using Apache in NCS Clusters

The following setup is required to use Apache HTTP Server on your OES servers in Novell Cluster Services clusters:

- You can use cluster-enabled Novell Storage Services (NSS) file system or Linux LVM volumes to host your web content:

  - **NSS volumes:** Install the **Novell Storage Services** pattern on each OES node in the cluster. For information, see "Installing and Configuring Novell Storage Services" in the *OES 2015 SP1: NSS File System Administration Guide for Linux*.

    Novell Cluster Services supports cluster resources for NSS pools. For information about creating clustered NSS pools and volumes, see "Configuring and Managing Cluster Resources for Shared NSS Pools and Volumes" in the *OES 2015 SP1: Novell Cluster Services for Linux Administration Guide*.

- **Linux LVM volumes:** Novell Cluster Services supports cluster resources for LVM volume groups and NCP-enabled LVM volume groups. For information about creating clustered LVM volume groups and volumes with and without NCP, see "Configuring and Managing Cluster Resources for Shared LVM Volume Groups" in the *OES 2015 SP1: Novell Cluster Services for Linux Administration Guide*.

  You can create and manage cluster-enabled LVM volumes with NSSMU and NLVM if you install Novell Storage Services on each node in the cluster.

- If you host multiple websites on a server in a Novell Cluster Services cluster, you must configure an Apache virtual host for each website on one OES node, then copy the configuration files to every OES node in the cluster.

- In a Novell Cluster Services cluster, the directories you specify in the `DocumentRoot` directive and any `Alias` directives for a virtual host should reside on the same cluster resource so they can fail over together. The location that contains the web content should be a directory on the volume, not the root of the volume. Specify the full Linux path of the directory. Linux paths are case-sensitive.

  For example, Novell Cluster Services scripts assume that the pool's volumes are mounted in the default NSS location of `/media/nss/<volume_name>`. The full Linux path of the `/www/mysite` path on an NSS volume `APACHEVOL` is

  `/media/nss/APACHEVOL/www/mysite`

  For LVM logical volumes, specify the mount point of the file systems. The default location for LVM volumes created with NSSMU or NLVM commands is `/usr/novell/<volume_name>`. The full Linux path of the `/www/mysite` path on an LVM logical volume `APACHEVOL` is

  `/usr/novell/APACHVOL/www/mysite`

4.3 Understanding the Default OES Setup of Apache HTTP Server

When you install services from the OES Add-On disk, the following Apache setup is configured:

- Section 4.3.1, "Apache and Tomcat Installation," on page 41
- Section 4.3.2, "Apache HTTP Server Configuration," on page 42
- Section 4.3.3, "Apache User wwwrun and Group www," on page 42
- Section 4.3.4, "Virtual Host for the OES Welcome Website," on page 42
- Section 4.3.5, "Secure SSL Virtual Host for the Default Website," on page 42
- Section 4.3.6, "Secure SSL Virtual Host for the Novell iManager Website," on page 43

4.3.1 Apache and Tomcat Installation

Novell-ready versions of Apache 2 HTTP Server software (Prefork, 64-bit) and Tomcat 6 are automatically installed when you set up OES services on a server.

4.3.2 Apache HTTP Server Configuration

OES configures Apache settings in the `/etc/sysconfig/apache2` global configuration file and the `/etc/apache2/conf.d/oes_httpd.conf` configuration file.

The `/etc/sysconfig/apache2` configuration file controls some global settings of Apache, such as modules to load, additional configuration files to include, server flags to apply when the Apache HTTP Server daemon (`httpd2`) is started, and flags that should be added to the command line.

4.3.3 Apache User wwwrun and Group www

Apache uses the user `wwwrun` identity to serve files to clients of your website. OES and Apache configure the following during the OES installation:

- The Apache installation creates a local group `www` and user `wwwrun` on the server. OES automatically makes the user `wwwrun` as the file owner of the web content for the OES Welcome website.

  When you create a website location, you must configure the user `wwwrun` as the file owner of the website's main directory and files.

- OES creates the group `www` and the user `wwwrun` in eDirectory when you install an OES server in an eDirectory tree for the first time. The user `wwwrun` is added as a member of the group `www`. The user `novlxsrvd` is also created and added to the group `www`.

- OES enables the group `www` and its member users (`wwwrun` and `novlxsrvd`) for Linux with Linux User Management (LUM).

  If your website is hosted on an NSS volume or an NCP-enabled Linux volume, you must assign the eDirectory user `wwwrun` as a file system trustee of the website's main directory, and give the trustee Read and File Scan rights.

For information about changing the file owner or configuring a file system trustee, see Section 4.4.6, "Configuring Permissions for the Website DocumentRoot Directory," on page 49.

4.3.4 Virtual Host for the OES Welcome Website

OES automatically configures the OES Welcome website in the `/etc/opt/novell/httpd/conf.d/welcome-apache.conf` file. Listening is set up on port 80 in the `/etc/apache2/listen.conf` file. Port 80 is opened in the firewall. The Apache HTTP Server daemon (`httpd2`) starts automatically on server restart.

Apache serves the Welcome page for the OES server at

http://<*server_dns_or_ip_address*>

4.3.5 Secure SSL Virtual Host for the Default Website

OES automatically configures a default secure virtual host (`_default_:443`) in the `/etc/apache2/vhost.d/vhost-ssl.conf` file. It sets up listening on port 443 in the `/etc/apache2/listen.conf` file. It opens port 443 in the firewall. The default virtual host configuration is automatically loaded first. It is also used when a domain name does not match a virtual host configuration. The default virtual host defines a custom log `/var/log/apache2/ssl_request_log` to capture events for SSL requests. An `Include` directive in the `/etc/apache2/vhost.d/vhost-ssl.conf` file automatically loads the virtual hosts that are defined in the `/etc/opt/novell/httpd/sslconf.d/*.conf` files.

4.3.6 Secure SSL Virtual Host for the Novell iManager Website

If you install Novell iManager on an OES server, the iManager installation automatically configures a secure virtual host for iManager and Novell Portal Services (NPS) in the `/etc/opt/novell/iManager/nps-Apache.conf` file. A symbolic link in the `/etc/opt/novell/httpd/sslconf.d/` directory points to the `nps-Apache.conf` file. This allows the virtual host to be automatically included along with the default secure virtual host when Apache is restarted.

Aliases are defined in the `nps-Apache.conf` file to hit the website with any of the following URLs:

https://<*server_dns_or_ip_address*>/nps/iManager.html

https://<*server_dns_or_ip_address*>/nps

https://<*server_dns_or_ip_address*>/iManager.html

4.4 Using Apache HTTP Server on OES Servers (Single Server or Cluster Nodes)

When you set up OES services on the server, Novell-ready versions of Apache 2 HTTP Server software (Prefork, 64-bit) and Tomcat 6 are automatically installed. Apache and the OES Welcome website are automatically configured for non-secure port 80 and secure port 443. The Apache HTTP Server daemon (`httpd2`) starts automatically on server restart. For more information, see Section 4.3, "Understanding the Default OES Setup of Apache HTTP Server," on page 41.

To set up personalized websites, you must manually create a virtual host configuration file for each website. Templates for secure SSL virtual host and non-secure virtual host configuration files are available in the `/etc/apache2/vhosts.d/` directory. Use a text editor to create or modify the configuration files, then gracefully restart the Apache HTTP Server daemon (`rcapache2 graceful`) to apply the changes.

WARNING: Do not use the **HTTP Server** option in YaST to configure Apache or virtual host settings on an OES server. It overwrites essential OES settings for Apache and breaks the existing setup. For recovery information, see Section 4.5.1, "Apache Server Errors after Using the HTTP Server Option in YaST," on page 54.

4.4.1 Configuring Apache Server Settings

On OES servers and Novell Open Workgroup Suite (NOWS) Small Business Edition (SBE) servers, you must manually configure Apache settings, OES virtual hosts, and virtual hosts for your personalized websites. Use a text editor to create or modify the configuration files, then gracefully restart the Apache HTTP Server daemon (`rcapache2 graceful`) to apply the changes.

WARNING: Do not use the **HTTP Server** option in YaST to manage Apache or the virtual host settings on an OES server. It overwrites essential OES settings for Apache and breaks the existing setup. For recovery information, see Section 4.5.1, "Apache Server Errors after Using the HTTP Server Option in YaST," on page 54.

For information about using the configuration files to manage your Apache HTTP Server and virtual hosts, see "Configuring Apache Manually" (http://www.suse.com/documentation/sles11/book_sle_admin/data/sec_apache2_configuration.html#sec_apache2_configuration_manually) in the *SLES 11 Administration Guide* (http://www.suse.com/documentation/sles11/book_sle_admin/data/book_sle_admin_pre.html).

4.4.2 Creating and Configuring a Virtual Host for Each Website

On Linux, the Apache HTTP server can serve multiple universal resource identifiers (URIs) from a single instance of Apache running on the server. That is, multiple websites, such as www.example.com and www.example.net, can be run from a single web server. Each website is referred to as a **virtual host**. Virtual hosts can be name based, IP based, or port based.

You can set up personalized websites by manually creating a virtual host configuration file for each website. Templates for secure SSL virtual host and non-secure virtual host configuration files are available in the `/etc/apache2/vhosts.d/` directory.

When you cluster-enable the web content by using Novell Cluster Services, use the IP address of the cluster resource for the virtual host. This ensures that the website traffic is directed to the cluster node where the web content cluster resource is currently active. Do not use the server node's IP address or the master IP address of the cluster. Specify the Linux file path to the web content. Keep in mind that Linux paths are case-sensitive.

On OES servers, you create and configure a separate virtual host configuration file for each website that you want to host in the cluster. The following procedure provides basic information about setting up the virtual host file. Refer to other sections in this document to learn about the key settings that are available. For detailed information, see the Apache Virtual Host documentation website (http://httpd.apache.org/docs/2.2/vhosts/).

IMPORTANT: The following procedure assumes that the website contents reside on a clustered NSS volume. If you use a clustered LVM volume or a clustered NCP-enabled LVM volume, modify the paths according to your configuration.

1 Choose an OES node in the cluster, then log in as the `root` user.

2 Create a copy of the virtual host template file in the `/etc/apache2/vhosts.d/` directory.

 The `/etc/apache2/vhosts.d/` directory contains a basic template (`vhost.template`) for a non-secure virtual host and an SSL template (`vhost-ssl.template`) for a secure virtual host.

3 Rename the file with a name for your virtual host, and add the `.conf` file extension, such as `mysite-Apache.conf`.

4 Open the virtual host file in a text editor and configure the virtual host settings for your personalized website:

4a If the web content is clustered with Novell Cluster Services, set the `VirtualHost` directive to the IP address or DNS host name assigned to the cluster resource:

```
<VirtualHost hostname>
```

For example, if the DNS name is `mysite.example.com`, specify `mysite` as the `VirtualHost`.

```
<VirtualHost mysite>
```

4b Set the value of the `DocumentRoot` directive to the Linux path of the directory where you placed your web content, and specify the directory options for this location.

The target directory must contain an `index.html` file, which is the root document for the virtual host. Specify the Linux path to the directory. For example, if you place your web content in an NSS volume path `APACHEVOL:\www\mysite`, the Linux path is `/media/nss/APACHEVOL/www/mysite`.

```
DocumentRoot "/media/nss/APACHEVOL/www/mysite"

<Directory  "/media/nss/APACHEVOL/www/mysite">
   # Possible options are "None", "All" or any combination of:
   # Indexes Includes FollowSymLinkx SymLinksifOwnerMatch ExecCGI MultiViews

   Options Indexes MultiViews
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>
```

4c Configure the host settings as desired for other directives in the file.

The minimum settings for a non-secure website are shown in the following example:

```
<VirtualHost mysite>

DocumentRoot "/media/nss/APACHEVOL/www/mysite"

ServerAdmin  mysite-admin@example.com
ServerName   mysite.example.com

ErrorLog     /var/log/apache2/error_log
TransferLog  /var/log/apache2/access_log
#CustomLog   /var/log/apache2/mysite.example.com-access_log combined


HostnameLookups On

UseCanonicalName On

ServerSignature Off

<Directory  "/media/nss/APACHEVOL/www/mysite">
   # Possible options are "None", "All" or any combination of:
   # Indexes Includes FollowSymLinkx SymLinksifOwnerMatch ExecCGI MultiViews

   Options Indexes MultiViews
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>

</VirtualHost>
```

4d (Optional) Specify alias paths in the virtual host configuration file.

For example, specify an alias for a Support web location that has a `support` directory at the same level as `mysite`. Include the `Alias` and `Directory` directives before the `</VirtualHost>` close tag.

```
Alias /support "/media/nss/APACHEVOL/www/support"
<Directory "media/nss/APACHVOL/www/support">
   Options Indexes MultiViews
   AllowOverride None
   Order deny,allow
   Allow from all
</Directory>
```

For information about alias paths that require LDAP authentication, see Section 4.4.7, "Configuring a Web Location that Requires LDAP Authentication," on page 51.

4e Save the virtual host configuration file.

5 (Optional) In the `/etc/apache2/listen.conf` file, add a `Listen` directive that specifies the IP address that you assigned to your cluster-enabled pool, and specify the port to use.

OES configures Apache to listen on non-secure port 80 by default. It listens for all traffic.

6 Make the websites visible on your network or to the world:

6a Add the site name and IP address resolution to your DNS server to make them visible.

6b If you use a non-standard port, open the port in the node's firewall.

6c If the traffic is from outside the firewall, open the port in the network firewall.

7 Gracefully restart the Apache HTTP Server daemon to apply the virtual host configuration:

```
rcapache2 graceful
```

Each `.conf` file is automatically included in the Apache configuration when you restart Apache.

8 Set up the virtual host for each of the remaining nodes:

8a Log in to the next node as the `root` user.

8b Copy the virtual host configuration file (such as `/etc/apache2/vhosts.d/mysite-apache.conf`) to the next node.

8c Create a local Linux path to the website that you specified in the `DocumentRoot` directive and to any paths you specified in `Alias` directives, then make the user `wwwrun` the owner of the directory and its contents.

When Apache is started or restarted, it looks for the paths specified in your website's virtual host configuration file. If a path does not exist, Apache reports an error but it loads the virtual host. Users access the site via the IP address or DNS name of the cluster resource, so web content is served only on the node where the resource is active.

When a cluster resource is not active on a node, the volume subdirectory (such as `APACHEVOL`) in the `/media/nss` directory is normally removed, and the path to the website does not exist. Creating the local path allows Apache to find the path even when the resource is not active on the node, and no error is reported when Apache loads. When the resource is taken offline, NSS does not remove the volume directory because it is now non-empty (it contains the local paths you create). The local path should not contain files. To add or remove web content files, access the NSS volume via the IP address of the cluster resource.

Enter the following commands for the website path and alias paths. The `chown` command changes the group to the Apache `www` group unless the group is the `root` user.

```
mkdir -p /media/nss/<volume_name>/<path>
```

```
chown wwwrun:www /media/nss/<volume_name>/<path>
```

For example, enter

```
mkdir -p /media/nss/APACHEVOL/www/mysite

chown wwwrun:www /media/nss/APACHEVOL/www/mysite

mkdir -p /media/nss/APACHEVOL/www/support

chown wwwrun:www /media/nss/APACHEVOL/www/support
```

8d Open a terminal console as the `root` user, then gracefully restart Apache:

```
rcapache2 graceful
```

8e Repeat these steps on each of the remaining nodes in turn.

IMPORTANT: Any time that you make changes to the virtual host configuration file, you must copy the modified file to every node in the cluster, and gracefully restart Apache on each node.

4.4.3 Requiring Strong Ciphers

We recommend that you secure your web solution by requiring strong ciphers when the client is negotiating the connection in the SSL handshake.

In OES 11 SP1 and later servers, the weak SSL ciphers are disabled by default in the `/etc/apache2/vhosts.d/vhost-ssl.conf` file:

```
# SSL Cipher Suite:
SSLCipherSuite ALL:!aNULL:!eNULL:!SSLv2:!LOW:!EXP:!MD5:@STRENGTH
```

On OES 11 and earlier servers, we recommend that you enable only the strongest ciphers: RSA, HIGH, and SSLv2.

To enable strong ciphers and disable weak ciphers in OES 11 and earlier:

1 In a text editor, modify the `/etc/apache2/vhosts.d/vhost-ssl.conf` file to require strong ciphers. Modify the default settings by placing a plus sign (+) before RSA, HIGH, and SSLv2, and placing an exclamation mark (!) before the weaker ciphers:

```
# SSL Cipher Suite:
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:!MEDIUM:!LOW:+SSLv2:!EXP:!eNULL
```

2 Gracefully restart Apache on the server:

```
rcapache2 graceful
```

3 Repeat this process on every Linux node in the cluster.

You can alternatively copy the Apache SSL configuration file (`/etc/apache2/vhosts.d/vhost-ssl.conf`) to every Linux node in the cluster, and then restart Apache.

For more information about SSL ciphers and cipher suites, see OpenSSL Ciphers (http://www.openssl.org/docs/apps/ciphers.html).

4.4.4 Configuring an SSL Certificate for the Server

OES automatically configures secure SSL communications for a default virtual host (`_default_:443`). SSL is enabled in the Apache global configuration file (`/etc/sysconfig/apache2`) with the following directive:

```
APACHE_SERVER_FLAGS="SSL"
```

The default SSL configuration is defined in the `/etc/apache2/vhosts.d/vhost-ssl.conf` file. It uses an `INCLUDE` directive for the `/etc/opt/novell/httpd/sslconf.d/*.conf` files. This target directory contains the configuration files (or symbolic links to them) for OES virtual hosts that require SSL, such as the `nps-Apache.conf` file that is used for the Novell iManager tool.

By default, OES sets up an SSL certificate file and key file for the server by using certificates generated in eDirectory. Table 4-1 identifies the location of the SSL certificate and key files that are referenced by the `SSLCertificateFile` and `SSLCertificateKeyFile` directives in the `/etc/apache2/vhosts.d/vhost-ssl.conf` file.

Table 4-1 *Default OES Server Certificates*

OES Server Certificate File	Location
SSL Certificate File	`/etc/ssl/servercerts/servercert.pem`
SSL Certificate Key File	`/etc/ssl/servercerts/serverkey.pem`

IMPORTANT: If you use SSL, set up a server certificate for each virtual host unless you use a wildcard certificate.

If you modify the content or location of the certificate and key files, gracefully restart the Apache HTTP Server daemon (`rcapache2 graceful`) to apply the new values.

4.4.5 Configuring Apache to Listen on Multiple Ports

The `Listen` directive in the `/etc/apache2/listen.conf` file tells the Apache HTTP Server to accept incoming requests on the specified port or an address-and-port combination. If the directive specifies only a port, the server listens to the given port on all interfaces. If the directive specifies an IP address and port combination, the server listens on the given port and network interface.

By default, OES configures Apache to listen on non-secure port 80 and secure port 443 in the `/etc/apache2/listen.conf` file. If a firewall is used on the server, port 80 and port 443 are automatically opened in the firewall. The ports are not bound to a particular IP address, so Apache responds to requests on all IP interfaces on the server.

```
Listen 80


<IfDefine SSL>
    <IfDefine !NOSSL>
  <IfModule mod_ssl.c>

      Listen 443

  </IfModule>
    </IfDefine>
</IfDefine>
```

You can configure multiple `Listen` directives to specify multiple IP addresses and ports. The server responds to requests from any of the listed addresses and ports. For information about formats and options for the `Listen` directive, see the *Listen Directive* (http://httpd.apache.org/docs/2.2/mod/mpm_common.html#listen) in the *Apache MPM Common Directives* collection.

If you configure non-standard ports for your personalized websites, you must add a `Listen` directive in the `/etc/apache2/listen.conf` file, then gracefully restart the Apache HTTP Server daemon (`rcapache2 graceful`) to apply the changes. Ensure that you open the port in the firewall.

4.4.6 Configuring Permissions for the Website DocumentRoot Directory

Apache uses the user `wwwrun` identity to serve files to clients of your website. You must configure permissions for the website content that allow Apache to serve the files to client users.

Setting the User wwwrun as the Owner of the Website's Directory and Files

The user `wwwrun` must be the file owner of the website's main directory and files.

1 Log in as the `root` user, then open a terminal console.

2 Use the change directory (`cd`) command to go to the directory that contains the main directory of your website. This is the directory you specify as the `DocumentRoot` in the virtual host configuration file.

For example, if the `DocumentRoot` is /media/nss/APACHEVOL/www/mysite, enter

`cd /media/nss/APACHEVOL/www`

3 Change the owner of the website's directory and files to user `wwwrun`. Enter:

`chown -R wwwrun:www mysite`

This recursively modifies the owner to user `wwwrun` for the directory and the subdirectories and files it contains. It changes the group to `www` unless the group is set to the `root` user.

4 In a file browser, view the directory's properties to verify that the owner was changed.
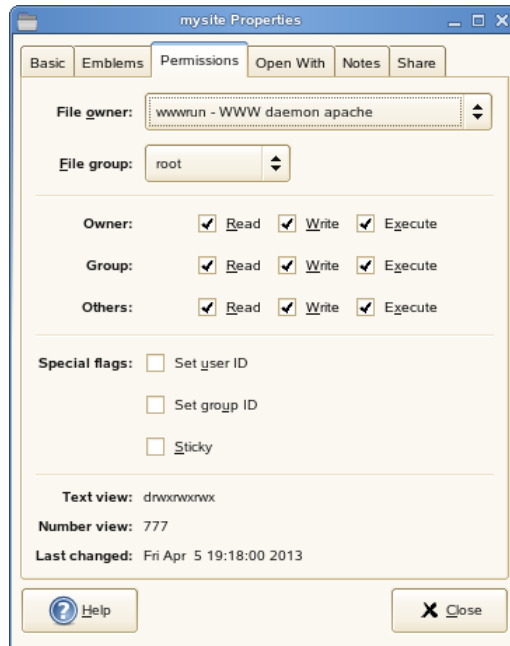
You can also use the `ls -al <path>` command to list the directory and view the owner, group, and permissions.

Setting User wwwrun as a File System Trustee of the Website's Directory

OES automatically creates the user `wwwrun` and group `www` in eDirectory. Both are LUM-enabled. You can verify their configuration by using the **Directory Administration** option and **Linux User Management** option in Novell iManager.

If your website is hosted on an NSS volume or an NCP-enabled Linux volume, you must assign the eDirectory user `wwwrun` as a file system trustee of the website's main directory, and give the trustee Read and File Scan rights. You can also set the `www` group as a trustee with Read and File Scan rights.

1 Log in to Novell iManager as an administrator user.

2 In the iManager toolbar, click the **View Objects** icon.

3 In the Tree view, select the volume, then browse the file system to locate the directory that contains your website's content.

4 Select the check box next to the directory, then select **Actions > Properties.**

5 On the Properties page, select **Rights**.

6 Click the **Add Trustee** browse icon to open the **Object Selector**.

7 Locate and select the user `wwwrun`, then click **OK**.

The user `wwwrun` is added as a trustee with the default Read and File Scan rights.
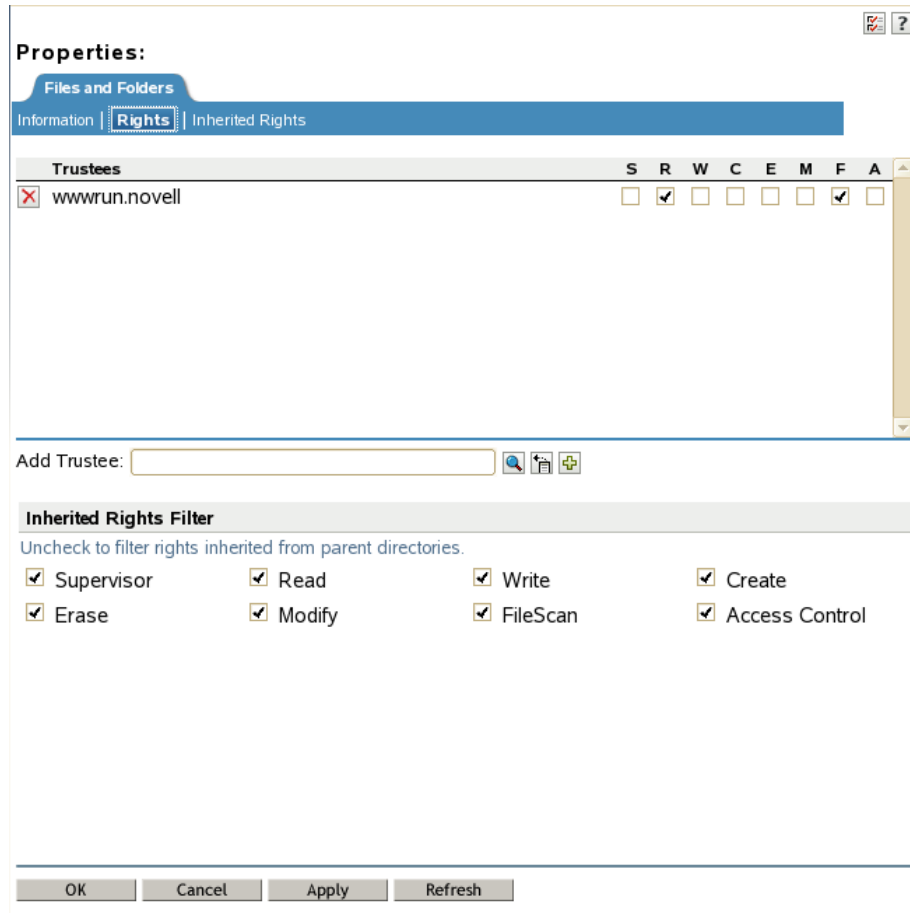
8 Click **Apply** or **OK** to save the changes.

4.4.7 Configuring a Web Location that Requires LDAP Authentication

If you have documents or a location that requires restricted web access, you can set up Apache to enforce eDirectory authentication and force the authentication to be done over `https`. This solution can be used on individual directories, URLs, or the entire Apache server.

The following example creates a single secure location so that any document that is referenced under the directory requires authentication. For example, the URL www.example.com can have public access, while the URL www.example.com/secure and documents it contains require authentication. Authentication should be done over a secure connection (https) rather than a non-secure connection (`http`). All `http` attempts are redirected to `https` for the given location.

1 Ensure that the `rewrite` module is enabled in the `/etc/sysconfig/apache2` global configuration file. OES enables this module by default.

Open the `/etc/sysconfig/apache2` file in a text editor, and verify that `rewrite` is listed in the modules defined in the `APACHE_MODULES` directive.

2 Configure the permissions for the user `wwwrun` on the target directory:

  2a Change the owner to the Apache user `wwwrun`:

```
chown -R wwwrun:www /media/nss/APACHEVOL/www/secure
```

This changes the group to the Apache group `www` unless the group is the `root` user.

2b For an NSS volume or an NCP-enabled Linux volume, configure the user `wwwrun` as a file system trustee of the `/media/nss/APACHEVOL/www/secure` directory, and give the trustee Read and File Scan rights.

For information, see "Setting User wwwrun as a File System Trustee of the Website's Directory" on page 50.

3 In a text editor, create a copy of the `/etc/apache2/vhosts.d/vhosts-ssl.template` file to create a `secure.conf` configuration file.

4 Allow for all `http` requests for the `/secure` alias to be redirected to `https`.

Add the following directives to the `secure.conf` file:

```
RewriteEngine On

RewriteRule ^/secure https://%{SERVER_NAME}/secure [L,R]
```

5 If the location that contains secure information exists outside the `DocumentRoot` directory, create an alias to the directory.

Add the following line to the `secure.conf` file:

```
Alias /secure "/<path_to_directory>/secure"
```

For a cluster resource, the secure directory ideally resides on the same clustered volume as the website, and at the same directory level as `DocumentRoot` for the website:

```
Alias /secure "/media/nss/APACHEVOL/www/secure"
```

6 Under the `Alias` directive, add the option for LDAP authentication under the `Directory` directive in the `secure.conf` file. Specify the IP address or DNS name of the website's cluster resource.

```
<Directory "media/nss/APACHVOL/www/secure">
  Options Indexes MultiViews
  AllowOverride None
  Order deny,allow
  Allow from all
  AuthType Basic
  AuthName "Protected"
  require valid-user
  AuthzLDAPAuthoritative On
  AuthLDAPURL ldaps://<cluster_resource_ip_address_or_dns_name>/o=corp?uid?sub
</directory>
```

7 Save the `/etc/apache/vhosts.d/secure.conf` file.

8 Open a terminal console as the `root` user, then gracefully restart the Apache daemon:

```
rcapache2 graceful
```

9 Verify that Apache is able to start.

If there are errors, make corrections in the configuration file, then restart the Apache daemon.

10 In a web browser, go to the website with `http` and verify that you are redirected to `https`, and that you can authenticate against the `/secure` alias.

4.4.8 Starting, Stopping, or Restarting the Apache Daemon

The Apache HTTP Server program runs as a daemon (`httpd2`) that executes continuously in the background to handle requests. OES configures the daemon to start automatically on system restart. You must restart Apache to apply any changes you make to the Apache or virtual host configuration files, or to add new virtual host configuration files. A graceful restart does not disrupt the service.

In a cluster, you manually copy the virtual host configuration files for clustered personalized websites to every node in the cluster. When Apache starts on each node, it reads the configuration file and is available to serve the site when the resource is active on the node. You do not add Apache commands in the resource's load and unload scripts. All requests to a clustered website are sent to the DNS name or IP address of the cluster resource, and not to a specific node. The site's requests are served by the Apache process that runs on the node where the cluster resource is currently active.

To start, stop, or restart the Apache daemon, use the `/usr/sbin/rcapache2` commands in Table 4-2:

Table 4-2   */usr/sbin Commands*

Command	Description
rcapache2 start	Starts the `httpd2` parent process. The parent process reads its configuration files and opens log files, and then spawns the child processes to serve hits.
	OES configures the Apache daemon to start automatically on server restart.
rcapache2 stop	Causes the parent process to immediately attempt to kill all of its child processes. This can take several seconds. The parent exits after all child processes have exited. Any requests in progress are terminated, and no further requests are served.
rcapache2 graceful-stop	Causes the parent process to advise its child processes to exit after their current request (or to exit immediately if they are not serving anything). The parent removes its PID file and ceases listening on all ports. The parent continues to run, and monitors child processes that are handling requests. The parent exits after the child processes complete the pending requests and exit, or when a timeout period has elapsed (as specified by the `GracefulShutdownTimeout`). If the timeout is reached, any remaining child processes are automatically sent the TERM signal to force them to exit, and any requests in progress are terminated.
rcapache2 restart	Causes the parent process to immediately kill its child processes such as the `stop` option, but the parent does not exit. It re-reads its configuration files, and re-opens any log files. Then it spawns a new set of child processes and continues serving hits.
rcapache2 graceful	Causes the parent process to advise the child processes to exit after their current request (or to exit immediately if they are not serving anything). The parent re-reads its configuration files and re-opens its log files. As each child dies, the parent replaces it with a child from the new generation of the configuration, which begins serving new requests immediately.

4.4.9 Viewing the Apache Log Files

The following Apache log files are located in the `/var/log/apache2/` directory:

```
access_log
error_log
rcapache2.out
rewrite_log
ssl_request_log
```

You can also specify custom logs by adding the `CustomLog` directive to your virtual host configuration file. For information about formatting the custom log, see *Apache Module mod_log_config* (http://httpd.apache.org/docs/2.2/mod/mod_log_config.html).

4.5 Troubleshooting the Apache HTTP Server

This section describes some issues you might experience with Apache HTTP Server and provides suggestions for resolving or avoiding them. For additional troubleshooting information, see the Novell Technical Support Knowledgebase (http://www.novell.com/support).

- Section 4.5.1, "Apache Server Errors after Using the HTTP Server Option in YaST," on page 54
- Section 4.5.2, "Files Downloaded from NetStorage Are 0 Bytes," on page 55

4.5.1 Apache Server Errors after Using the HTTP Server Option in YaST

If you use the **HTTP Server** option in YaST to manage Apache or virtual hosts, the option can overwrite essential OES settings and load the wrong modules, which breaks the default Apache HTTP Server setup. For information, see TID 7002562 (http://www.novell.com/support/kb/doc.php?id=7002562) in the Novell Knowledgebase.

If you have used the **HTTP Server** option in YaST and Apache is no longer working, recover the OES default Apache HTTP Server setup by doing the following:

1 As the `root` user, open the `/etc/sysconfig/apache2` file in a text editor and modify the following directives:

- **Proxy module:** In the `APACHE_MODULES=` line in the file, ensure that the `proxy` module is listed before the `proxy_ajp` module. For example (some modules are not listed for ease of reading the example):

  `APACHE_MODULES="cgi dir rewrite ssl proxy proxy_ajp ssl"`

- **SSL module:** In the `APACHE_MODULES=` line in the file, ensure that the `ssl` module is listed. For example (some modules are not listed for ease of reading the example):

  `APACHE_MODULES="cgi dir rewrite ssl proxy proxy_ajp ssl"`

- **Prefork mode:** Apache should run in `prefork` mode rather than `worker` mode. To force this, ensure that the `APACHE_MPM=""` line is set to `"prefork"`. For example:

  `APACHE_MPM="prefork"`

- **SSL:** Ensure secure communications by enabling the SSL flag. For example:

  `APACHE_SERVER_FLAGS="SSL"`

2 Gracefully restart Apache to apply the changes. As the `root` user, enter the following command at a console prompt:

```
rcapache2 graceful
```

4.5.2 Files Downloaded from NetStorage Are 0 Bytes

After you lock down ciphers for an Apache HTTP Server to use only the strongest SSL ciphers, all of the files downloaded from NetStorage are 0 bytes in size.

NetStorage might not work as expected if you lock down Apache HTTP Server to disallow low and medium SSL ciphers. Try allowing medium SSL cipher settings to see if that is sufficient, then add back low cipher settings if necessary.

For other SSL cipher configuration options, see *SSL/TLS Strong Encryption: How-To* (http://httpd.apache.org/docs/2.2/ssl/ssl_howto.html) at Apache.org.

4.6 Additional Information

The latest Apache documentation is available on the Apache HTTP Server Version 2.2 Documentation website (http://httpd.apache.org/docs-2.2/).

A SLES SP3 Modifications

Upgrading from MySQL 5.0 to MySQL 5.5 Introduces a New Database Format

SUSE Linux Enterprise Server 11 SP3 introduces an upgrade of MySQL from version 5.0 to version 5.5. This upgrade involves a change of the database format. You must manually migrate your existing MySQL database to the new format before MySQL can run again. After you upgrade to MySQL 5.5, its daemon is not automatically started. We recommend that you back up the database before you migrate it to the new format.

To migrate an existing MySQL database to the version 5.5 database format:

1 Log in to the server as the *root* user, then open a terminal console.

2 At the command prompt, enter

```
touch /var/lib/mysql/.force_upgrade
```

This assumes a local database setup path.

If the database is clustered with Novell Cluster Services, take the MySQL cluster resource offline on a node, manually mount the volume locally, then issue the command using the path to the location on the mounted volume. For example:

```
touch /mnt/mysql/var/lib/mysql/.force_upgrade
```

After the database is reformatted, dismount the volume locally.

3 After the migration to the new database format is complete, start MySQL. At the command prompt, enter

```
rcmysql restart
```

4 If the database is clustered with Novell Cluster Services, bring the MySQL cluster resource online.

Upgrading from PostgreSQL 8.3 to PostgreSQL 9.1 Introduces a New Database Format

SLES 11 SP3 introduces an upgrade of PostgreSQL from version 8.3 to version 9.1. This upgrade involves a change of the database format. You must manually migrate your existing PostgreSQL database to the new format before PostgreSQL can run again. After you upgrade to PostgreSQL 9.1, its daemon is not automatically started. We recommend that you back up the database before you migrate it to the new format.

A new `pg_upgrade` tool is provided to migrate the PostgreSQL database to the new format. Both the 8.3 version and 9.1 version of the software are included in SLES 11 SP3 to accommodate the use of this tool. For information about how to perform a database migration using the pg_upgrade tool, see the `pg_upgrade` tool documentation (`/usr/share/doc/packages/postgresql91/html/pgupgrade.html`) on the server (requires the `postgresql91-docs` package).

1 Ensure that you apply the latest patches for PostgreSQL 8.3.

The latest patch relocates the software from its standard location to a versioned location such as `/usr/lib/postgresql83/bin`. Symbolic links are used to make the software available in the standard location.

2 Install PostgreSQL 9.1 and its dependent packages. The `pg_upgrade` tool is found in the `postgresql91-contrib` package.

The packages are installed to a versioned location such as `/usr/lib/postgresql91/bin`. Symbolic links are used to make the software available in the standard location instead of version 8.3.

3 Use the `pg_upgrade` tool to migrate the PostgreSQL database format from version 8.3 to version 9.1.

Unless the tool is used in link mode, the server must have enough free disk space to temporarily hold a copy of the database files. You can run the `du -hs` command to determine if enough space is available:

```
du -hs /var/lib/pgsql/data
```

This directory is the default PostgreSQL location for databases. Use the actual database path for your system.

PHP 5.3 Replaces PHP 5.2

PHP 5.3 is supported in SLES 11 SP3. The PHP 5.2 package has been removed. Both PHP 5.2 (deprecated) and PHP 5.3 were available in SLES 11 SP2.

WebSphere Application Server CE Is Removed in SLES 11 SP3

The WebSphere Application Server CE package has been removed from SLES 11 SP3. It is no longer supported.