

Web Services and Applications Guide

Novell® Open Enterprise Server 11

December 8, 2011

www.novell.com



Legal Notices

Novell, Inc., makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc., makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the [Novell International Trade Services Web page \(http://www.novell.com/info/exports/\)](http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Copyright © 2005 – 2011 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.
www.novell.com

Online Documentation: To access the latest online documentation for this and other Novell products, see the [Novell Documentation Web page \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

Novell Trademarks

For Novell trademarks, see [the Novell Trademark and Service Mark list \(http://www.novell.com/company/legal/trademarks/tmlist.html\)](http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

This product includes the following open source products that are discussed in this guide:

Apache

Apache HTTP Server 2.2 provides an HTTP/1.1 compliant Web server. It is an open source software product developed by the [Apache Software Foundation \(http://www.apache.org/\)](http://www.apache.org/). It is licensed under the [Apache License, Version 2.0 \(http://httpd.apache.org/docs/2.2/license.html\)](http://httpd.apache.org/docs/2.2/license.html).

Java

Java 6 Runtime Environment (java-1_6_0-ibm) contains the Java virtual machine, runtime class libraries, and Java application launcher that are necessary to run programs written in the Java programming language. It is an open source software product developed by [IBM \(http://www.ibm.com/developerworks/java/jdk/\)](http://www.ibm.com/developerworks/java/jdk/). It is licensed under the [IBM International License Agreement for Non-Warranted Programs \(ILAN\) \(http://www-03.ibm.com/software/sla/sladb.nsf/viewbla/\)](http://www-03.ibm.com/software/sla/sladb.nsf/viewbla/).

Java 2 Runtime Environment (`java-1_6_0-ibm-fonts`) contains the Java virtual machine, runtime class libraries, and Java application launcher that are necessary to run programs written in the Java programming language. It is an open source software product developed by [IBM](http://www-128.ibm.com/developerworks/java/jdk/) (<http://www-128.ibm.com/developerworks/java/jdk/>). It is licensed under the [IBM International License Agreement for Non-Warranted Programs \(ILAN\)](http://www-03.ibm.com/software/sla/sladb.nsf/viewbla/) (<http://www-03.ibm.com/software/sla/sladb.nsf/viewbla/>).

MySQL

MySQL 5.0.67 is a multiuser, multi-threaded, client/server implementation of the SQL database server. It is an open source software product developed by the [MySQL Community](https://www.mysql.com/products/community/) (<https://www.mysql.com/products/community/>). It is licensed under the [GNU General Public License, Version 2](http://opensource.org/licenses/gpl-2.0.php) (<http://opensource.org/licenses/gpl-2.0.php>).

Perl

Perl (Practical Extraction and Report Language) 5.10.0 is a programming language that is optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. It is an open source software product developed by [Perl.org](http://dev.perl.org/perl5/) (<http://dev.perl.org/perl5/>). It is licensed under the [Artistic License](http://dev.perl.org/licenses/artistic.html) (<http://dev.perl.org/licenses/artistic.html>) and the [GNU General Public License, Version 2](http://opensource.org/licenses/gpl-2.0.php) (<http://opensource.org/licenses/gpl-2.0.php>) or later version.

Embedded Perl Module 2.0.4 for Apache 2.0 (`apache2-mod_perl`) embeds a Perl (Practical Extraction and Report Language) interpreter in your Apache HTTP Server. It is an open source software product developed by the [Apache Software Foundation](http://www.apache.org/) (<http://www.apache.org/>). It is licensed under the [Apache Software License, Version 1.1](http://perl.apache.org/start/LICENSE.txt) (<http://perl.apache.org/start/LICENSE.txt>).

PHP 5

PHP 5 is a general-purpose scripting language that is well-suited for Web development and can be embedded in HTML. It is an open source software product developed by the [PHP Group](http://www.php.net/) (<http://www.php.net/>). It is licensed under the [PHP License, version 3.1](http://www.php.net/license/3_01.txt) (http://www.php.net/license/3_01.txt).

PHP 5 Extension Module 5.2.6 for MySQL (`php5_mysql`) provides PHP functions for access to your MySQL database server. It is an open source software product developed by the [PHP Group](http://www.php.net/) (<http://www.php.net/>). It is licensed under the [PHP License, version 3.1](http://www.php.net/license/3_01.txt) (http://www.php.net/license/3_01.txt).

PHP 5 Module 5.2.6 for Apache 2.0 (`apache2-mod_php5`) is a server-side, cross-platform HTML embedded scripting language for your Apache HTTP Server. It is an open source software product developed by the [PHP Group](http://www.php.net/) (<http://www.php.net/>). It is licensed under the [PHP License, Version 3.1](http://www.php.net/license/3_01.txt) (http://www.php.net/license/3_01.txt).

PostgreSQL

PostgreSQL 8.3.9 is an 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. It is an open source project developed by the [PostgreSQL Global Development Group](http://www.postgresql.org/) (<http://www.postgresql.org/>). It is licensed under the [PostgreSQL License](http://www.postgresql.org/about/licence) (<http://www.postgresql.org/about/licence>), a liberal Open Source license, similar to the [BSD 3-Clause License](http://www.opensource.org/licenses/BSD-3-Clause) (<http://www.opensource.org/licenses/BSD-3-Clause>).

Python

Python 2.6.0 is an interpreted, object-oriented programming language. It is an open source software product developed by the [Python Software Foundation](http://python.org/) (<http://python.org/>). It is licensed under the [MIT X11 License](http://www.opensource.org/licenses/mit-license.php) (<http://www.opensource.org/licenses/mit-license.php>).

Python Module 1.2.2 for MySQL (`python-mysql`) provides Python functions for access to your MySQL database server. It is an open source software product developed by the [MySQL-Python project on Sourceforge.net](http://sourceforge.net/projects/mysql-python/) (<http://sourceforge.net/projects/mysql-python/>). It is licensed under the [GNU General Public License, Version 2](http://opensource.org/licenses/gpl-2.0.php) (<http://opensource.org/licenses/gpl-2.0.php>) or later version.

Python Module 3.3.1 for Apache 2.0 (apache2-mod_python) embeds a Python interpreter in your Apache HTTP Server, which allows Apache to execute the Python code in-process. It is an open source software product developed by the [Apache Software Foundation](http://www.apache.org/) (<http://www.apache.org/>). It is licensed under the [Apache Software License, Version 1.1](http://perl.apache.org/start/LICENSE.txt) (<http://perl.apache.org/start/LICENSE.txt>) and [MIT X11 License](http://www.opensource.org/licenses/mit-license.php) (<http://www.opensource.org/licenses/mit-license.php>).

Tomcat

Tomcat 6.0.18 provides an open source software implementation of the Java Servlet and JavaServer Pages specifications that were developed by Sun under the Java Community Process. It is an open source software product developed by the [Apache Software Foundation](http://www.apache.org/) (<http://www.apache.org/>). It is licensed under the [Apache License, Version 2.0](http://httpd.apache.org/docs/2.2/license.html) (<http://httpd.apache.org/docs/2.2/license.html>).

WebSphere Application Server Community Edition

WebSphere Application Server Community Edition (CE) allows you to create less critical, departmental-level applications on a lower-cost, open source Java Enterprise Edition (Java EE) application server. It is an open source software product developed by [IBM](http://www-128.ibm.com/developerworks/java/jdk/) (<http://www-128.ibm.com/developerworks/java/jdk/>). It is licensed under the [IBM International License Agreement for Non-Warranted Programs \(ILAN\)](http://www-03.ibm.com/software/sla/sladb.nsf/viewbla/) (<http://www-03.ibm.com/software/sla/sladb.nsf/viewbla/>).

Contents

About This Guide	7
1 Overview of Web Services and Applications	9
1.1 Introduction to Web Services and Applications	9
1.1.1 What Are Web Services?	10
1.1.2 What Are Web Applications?	10
1.1.3 Web Application Tools (Java and J2EE)	11
1.1.4 Enabling Technologies	11
1.1.5 General Web Services and Applications Architecture	13
1.2 OES Components That Provide Web Services and Applications	13
1.2.1 Web Hosting: Apache Web Server	15
1.2.2 Servlet Support: Tomcat Servlet Container	16
1.2.3 Scripting: PHP and Perl	16
1.2.4 Web Databases: MySQL	16
1.2.5 Web Databases: PostgreSQL	17
1.2.6 Custom Web/J2EE Applications: WebSphere Application Server CE	18
1.2.7 Web and Network Search Capability: QuickFinder Server	19
1.3 What's Next	20
2 What's New or Changed in OES Web Services	21
3 Deploying Web Service and Application Components	23
3.1 Setting Up a Simple Web Server on Linux	23
3.2 Installing Web Services and Applications Packages in YaST	24
3.3 Managing Your Web Server on Linux	26
3.4 Additional Information	27
4 Configuring MySQL with Novell Cluster Services	29
4.1 Prerequisites for Clustering MySQL	30
4.2 Installing and Enabling MySQL	30
4.3 Creating an LVM Volume Group and Logical Volume	33
4.3.1 Sample Values	33
4.3.2 Setting Up the VG and LV (Overview)	33
4.3.3 Setting up the VG and LV (Detailed)	34
4.4 Configuring MySQL on the LVM Logical Volume	38
4.5 Cluster-Enabling MySQL on the Logical Volume	39
4.6 File Location	42
4.7 Security Considerations for the MySQL Configuration	43
4.7.1 MySQL Ports	43
4.7.2 Securing MySQL	43
4.8 Additional Information	44

About This Guide

Novell Open Enterprise Server (OES) 11 includes a collection of open source Web services and applications that let you build, deploy, host, and use Web sites 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 11.

The guide is divided into the following sections:

- ♦ [Chapter 1, “Overview of Web Services and Applications,” on page 9](#)
- ♦ [Chapter 2, “What’s New or Changed in OES Web Services,” on page 21](#)
- ♦ [Chapter 3, “Deploying Web Service and Application Components,” on page 23](#)
- ♦ [Chapter 4, “Configuring MySQL with Novell Cluster Services,” on page 29](#)

Audience

This guide is intended for Web or network administrators that install and manage Web site 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 11 Documentation Web site \(http://www.novell.com/documentation/oes11\)](http://www.novell.com/documentation/oes11).

Additional Documentation

For information about installing and using Apache 2.2 on SLES 11, see “[Apache HTTP Server](http://www.suse.com/documentation/sles11/book_sle_admin/data/cha_apache2.html)” (http://www.suse.com/documentation/sles11/book_sle_admin/data/cha_apache2.html) in the *SLES 11 Administration Guide* (http://www.suse.com/documentation/sles11/book_sle_admin/data/book_sle_admin_pre.html).

Each Linux 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 \(http://httpd.apache.org/docs/2.2/\)](http://httpd.apache.org/docs/2.2/)
- ♦ [Tomcat 6 documentation \(http://tomcat.apache.org/tomcat-6.0-doc/index.html\)](http://tomcat.apache.org/tomcat-6.0-doc/index.html)
- ♦ [MySQL documentation \(http://dev.mysql.com/doc\)](http://dev.mysql.com/doc)
- ♦ [PostgreSQL documentation \(http://www.postgresql.org/docs/\)](http://www.postgresql.org/docs/)
- ♦ [PHP Hypertext Preprocessor documentation \(http://www.php.net/docs.php\)](http://www.php.net/docs.php)
- ♦ [Perl documentation \(http://www.perl.org/docs.html\)](http://www.perl.org/docs.html)

- ♦ Python documentation (<http://www.python.org/doc/>)
- ♦ Ruby on Rails documentation (<http://rubyonrails.org/documentation>)

Overview of Web Services and Applications

1

Novell Open Enterprise Server (OES) 11 includes a collection of open source Web application services products that let you build, deploy, host, and use Web sites 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

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

This section covers the following topics:

- ♦ [Section 1.1, “Introduction to Web Services and Applications,” on page 9](#)
- ♦ [Section 1.2, “OES Components That Provide Web Services and Applications,” on page 13](#)
- ♦ [Section 1.3, “What’s Next,” on page 20](#)

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.

- ♦ [Section 1.1.1, “What Are Web Services?,” on page 10](#)
- ♦ [Section 1.1.2, “What Are Web Applications?,” on page 10](#)
- ♦ [Section 1.1.3, “Web Application Tools \(Java and J2EE\),” on page 11](#)
- ♦ [Section 1.1.4, “Enabling Technologies,” on page 11](#)
- ♦ [Section 1.1.5, “General Web Services and Applications Architecture,” on page 13](#)

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 Web site 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 Web site, 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” on page 12](#)
- ♦ [“Servlet Containers” on page 12](#)
- ♦ [“Web Database Servers” on page 12](#)
- ♦ [“Application Servers” on page 12](#)

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 Web sites and Web applications.

Application Servers

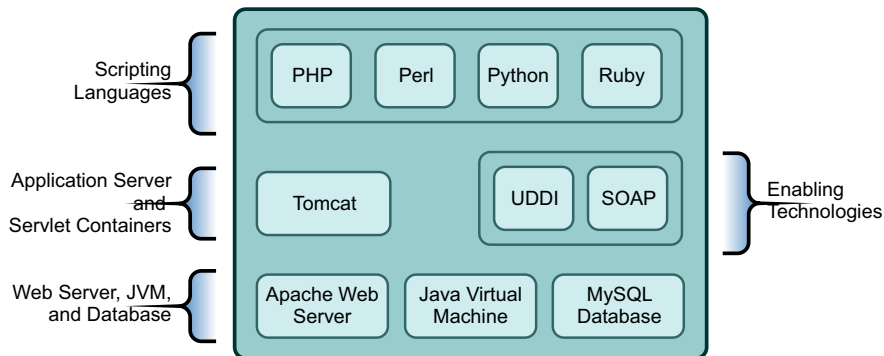
In more sophisticated Web application models, an application server is added to enable the system to manage business logic and track the user's progress through the application. The application server software runs in a middle tier, between Web browser-based clients and back-end databases and business 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 Web sites and build Web applications.

Figure 1-1 *Architecture of Key Web Components and Technologies*

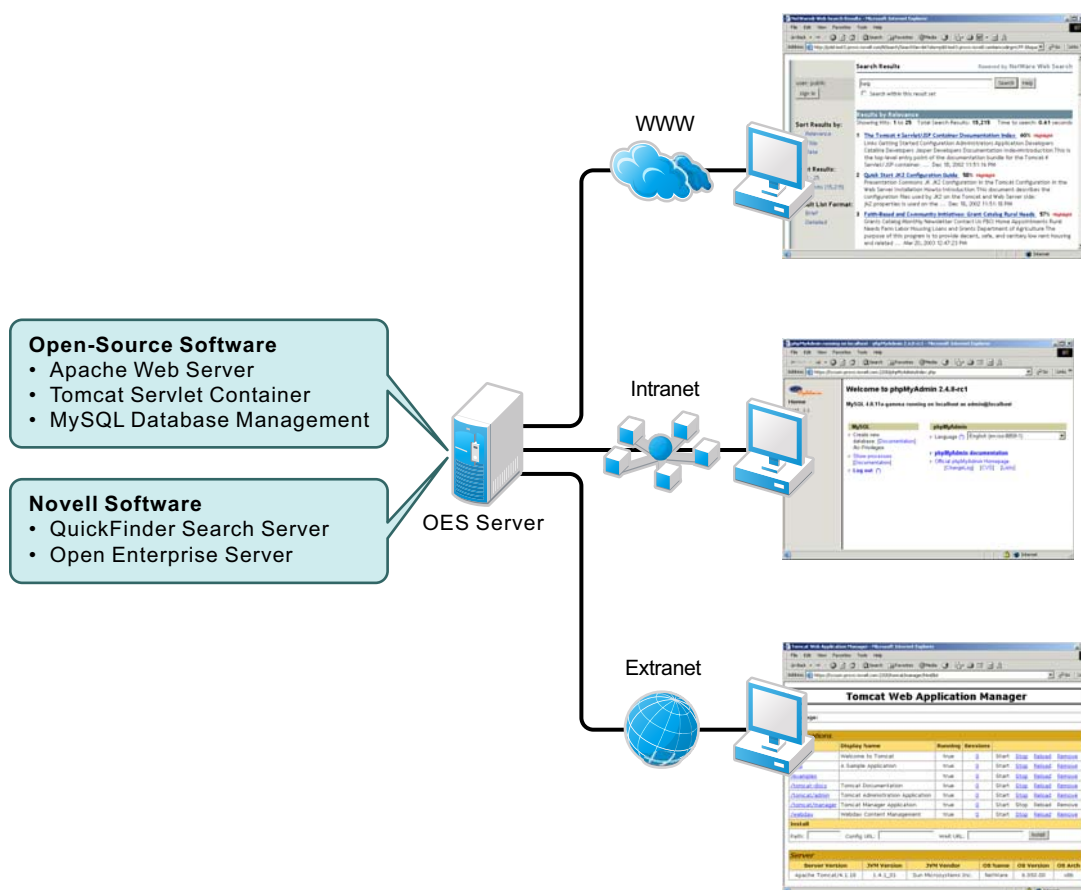


1.2 OES Components That Provide Web Services and Applications

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 Web sites 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 Web site, 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:

- ♦ [Section 1.2.1, “Web Hosting: Apache Web Server,” on page 15](#)
- ♦ [Section 1.2.2, “Servlet Support: Tomcat Servlet Container,” on page 16](#)
- ♦ [Section 1.2.3, “Scripting: PHP and Perl,” on page 16](#)
- ♦ [Section 1.2.4, “Web Databases: MySQL,” on page 16](#)
- ♦ [Section 1.2.5, “Web Databases: PostgreSQL,” on page 17](#)
- ♦ [Section 1.2.6, “Custom Web/J2EE Applications: WebSphere Application Server CE,” on page 18](#)
- ♦ [Section 1.2.7, “Web and Network Search Capability: QuickFinder Server,” on page 19](#)

1.2.1 Web Hosting: Apache Web Server

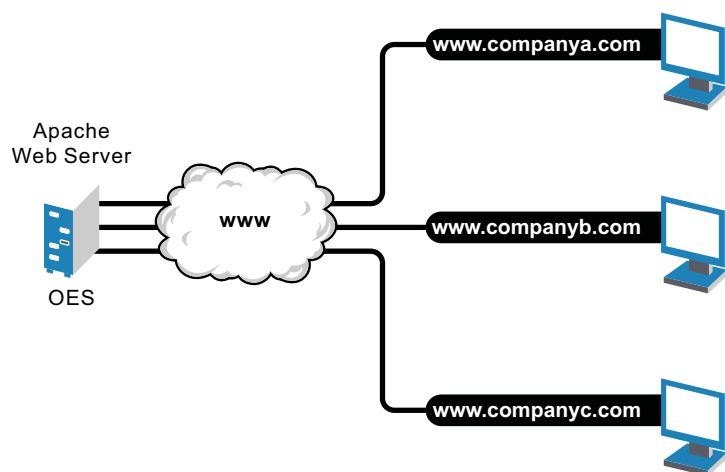
Apache is the most popular Web server being used on the World Wide Web today. Its popularity comes from the fact that it is the most reliable and secure Web server available. It runs on all major platforms, is capable of hosting even the most complex Web sites, and can scale to handle thousands of simultaneous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

The Apache Web Server serves as the foundation Web server upon which you can build Web sites 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 Web sites.
- ♦ 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 Web sites (with their own domain names) all from a single installation of Apache.

Figure 1-3 Apache Running on an OES Server and Hosting Multiple Web Site



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.

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.

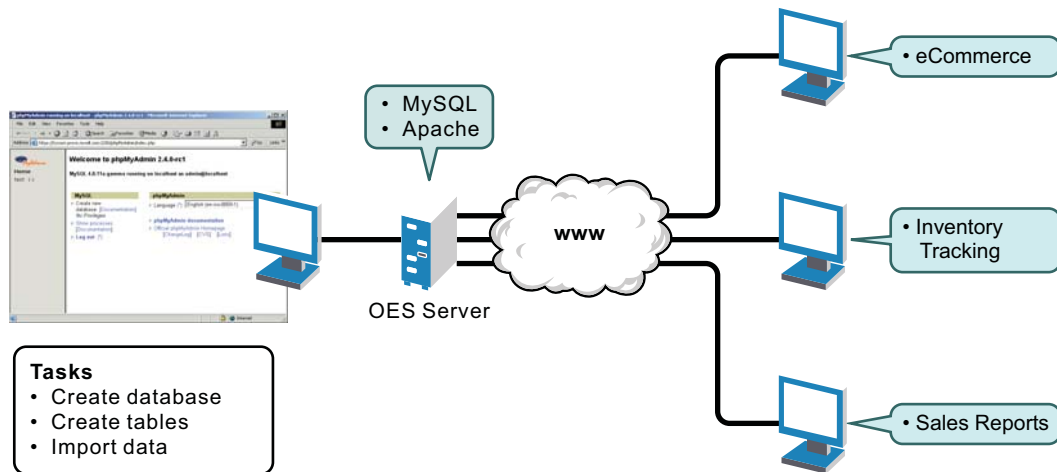
1.2.4 Web Databases: MySQL

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: The more powerful PostgreSQL database server comes with SUSE Linux Enterprise Server 9 and later.

1.2.5 Web Databases: PostgreSQL

SUSE Linux Enterprise Server 11 includes the open source PostgreSQL 8.3.9 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 than 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
<code>postgresql-server</code>	This package includes the programs needed to create and run a PostgreSQL server. It allows you to create and maintain PostgreSQL databases.
<code>postgresql</code>	This package contains the basic utility and client programs necessary to maintain and work with local or remote PostgreSQL databases and the <code>postgres(1)</code> , <code>initdb(1)</code> , <code>pg_ctl(1)</code> 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
<code>postgres</code>	The <code>postgres</code> command is used to manage the PostgreSQL database server instance.
<code>initdb</code>	The <code>initdb</code> command is used to create a new PostgreSQL database.
<code>pg_ctl</code>	The <code>pg_ctl</code> 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 8.3.15 Documentation \(http://www.postgresql.org/docs/8.3/interactive/index.html\)](http://www.postgresql.org/docs/8.3/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 2.1.1. Bundled with SLES 11, WebServer provides a J2EE-certified platform for building and deploying enterprise-class Web applications.

IBM WebSphere Application Server Community Edition 2.1.1 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 \(http://publib.boulder.ibm.com/wasce/Front_en.html\)](http://publib.boulder.ibm.com/wasce/Front_en.html).

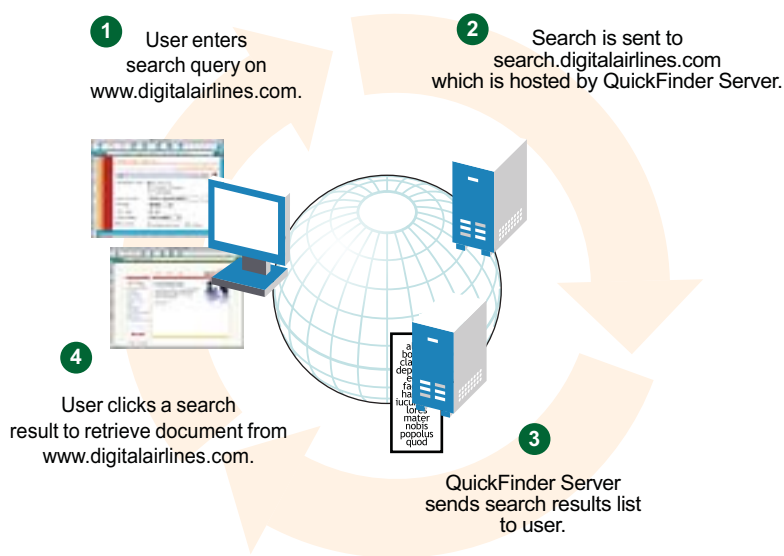
1.2.7 Web and Network Search Capability: QuickFinder Server

No Web solution is complete without capable searching functionality that provides users with a method for finding information they need, when they need it. That is why OES includes the Novell QuickFinder Server on the Linux platform.

Novell QuickFinder Server lets you add search and print functionality to any Web site, anywhere on the World Wide Web or on a company intranet. You can use it on your own enterprise-wide Web site or to host search services for business partners or clients.

Visitors to your Web or intranet site enter search terms in the search form that you place on the pages of your Web site. The search term is used to find matches contained in indexes you create using the QuickFinder Server Manager, a Web-based management utility. Search results, including matching URLs, are sent back to the user's Web browser.

Figure 1-5 How QuickFinder Server Handles a User's Search Query



For information about installing, configuring, and managing a QuickFinder Server, see the *OES 11: Novell QuickFinder Server 5.0 Administration Guide*.

1.3 What's Next

- ♦ To learn more about developing Web applications for the OES environment, see the [Novell Developer Web site \(http://developer.novell.com\)](http://developer.novell.com).
- ♦ For general OES installation instructions for Linux, see the *OES 11: Installation Guide*.

What's New or Changed in OES Web Services

2

The Web services and applications in Novell Open Enterprise Server (OES) 11 are Novell software and open source software that support SUSE Linux Enterprise Server (SLES) 11 Service Pack 1 (SP1).

Deploying Web Service and Application Components

3

Novell Open Enterprise Server (OES) components that provide Web application services on Linux can be installed during installation of the operating system, or they can be installed afterwards using YaST.

- ♦ [Section 3.1, “Setting Up a Simple Web Server on Linux,” on page 23](#)
- ♦ [Section 3.2, “Installing Web Services and Applications Packages in YaST,” on page 24](#)
- ♦ [Section 3.3, “Managing Your Web Server on Linux,” on page 26](#)
- ♦ [Section 3.4, “Additional Information,” on page 27](#)

3.1 Setting Up a Simple Web Server on Linux

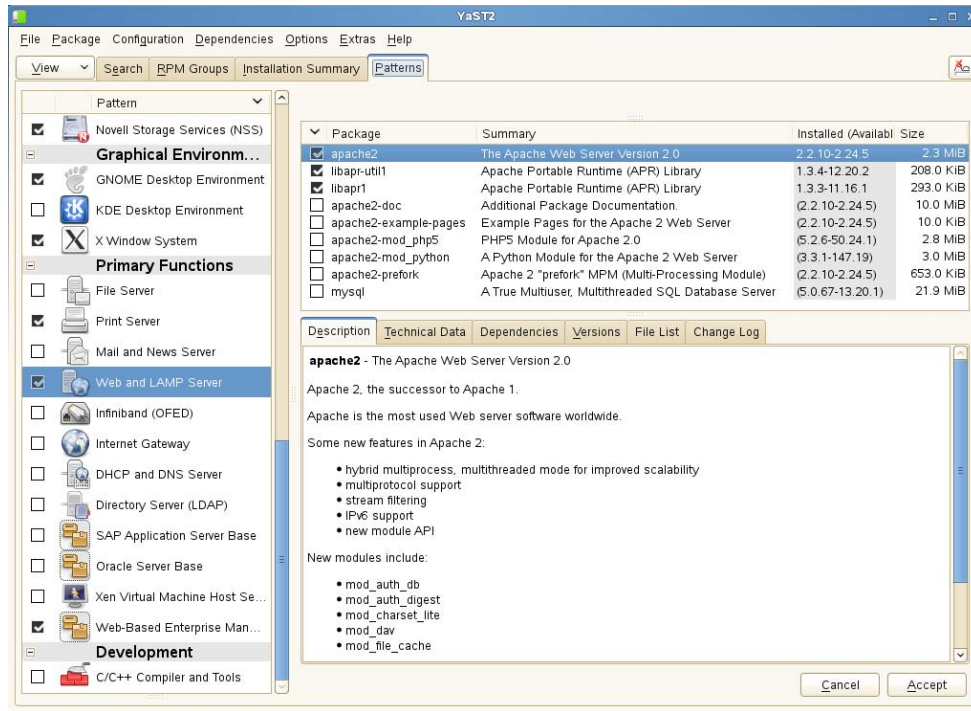
You can install the Web server software during the install or afterwards by selecting the Web and Lamp Server option under the Linux Primary Functions category. This provides software to set up a Web server that is able to serve static, dynamic, and interactive content (like a Web shop). The Web and LAMP Server pattern includes the Apache HTTP Server, the database management system MySQL, and scripting languages such as PHP, Python, Ruby on Rails, and Perl.

For complete instructions on installing Linux services on OES, see [OES 11: Installation Guide](#). For instructions on configuring Linux services, see the [SLES 11 Deployment Guide \(http://www.suse.com/documentation/sles11/book_sle_deployment/data/pre_sle.html\)](http://www.suse.com/documentation/sles11/book_sle_deployment/data/pre_sle.html).

To install the Web and LAMP Server pattern on an existing OES 11 server:

- 1 Log in as the `root` user, then open YaST.
- 2 In YaST, select *Open Enterprise Server > OES Install and Configuration*.
- 3 On the Software Selection page, under *Primary Functions*, select *Web and LAMP Server*, then click *Accept*.

If the software is already installed, a blue check box is displayed.



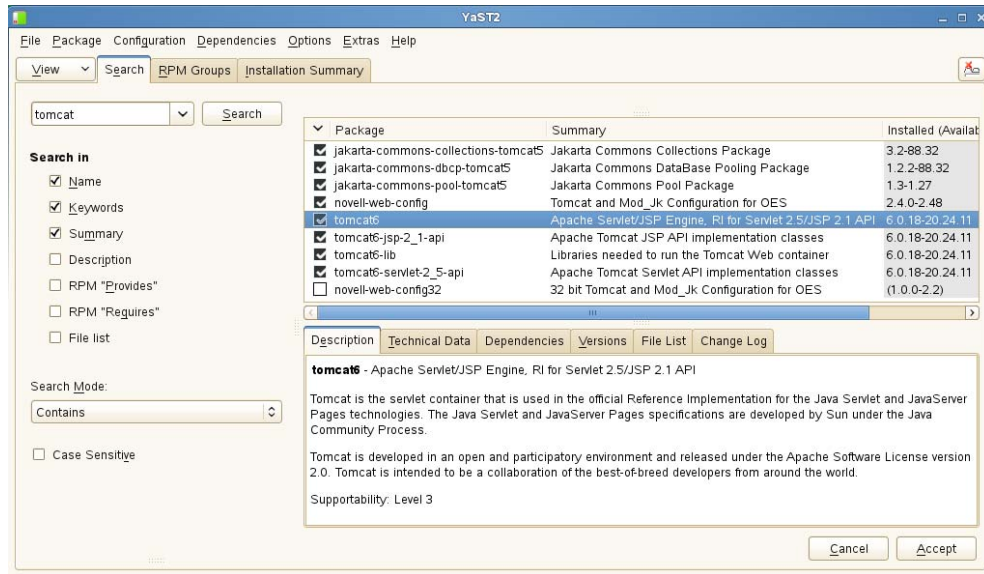
- 4 Continue with the installation.
- 5 After the install, configure the Apache Web Server.

For information, see “Apache HTTP Server” (http://www.suse.com/documentation/sles11/book_sle_admin/data/cha_apache2.html) in the *SLES 11 Administration Guide* (http://www.suse.com/documentation/sles11/book_sle_admin/data/book_sle_admin_pre.html).

3.2 Installing Web Services and Applications Packages in YaST

After you have installed Linux, you can use YaST to add more Web Services and Applications to your system. For example, to install Tomcat:

- 1 In YaST, click *Software* in the left pane, then click *Software Management* in the right pane.
- 2 In the Search field, type `tomcat`, then click *Search*.



3 Select the Tomcat packages you want to install, then click Accept.

If you encounter a dependency conflict, select the package you want to install to resolve the conflict, then click *OK -- Try Again* to continue with the installation.

Use this process to add Web Services and Applications packages to your Linux server as needed.

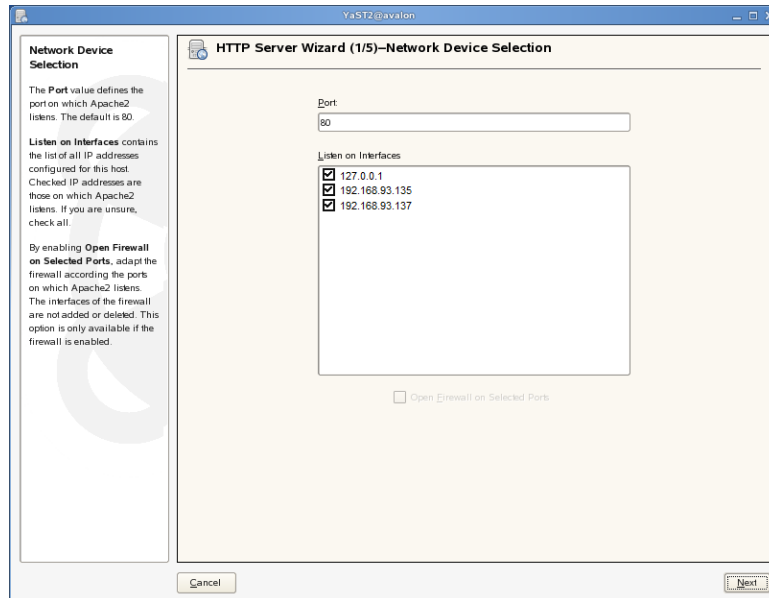
For more information about YaST, see “Installation with YaST” (http://www.suse.com/documentation/sles11/book_sle_deployment/data/cha_inst.html) in the *SLES 11 Deployment Guide* (http://www.suse.com/documentation/sles11/book_sle_deployment/data/pre_sle.html).

3.3 Managing Your Web Server on Linux

After you have installed and set up your Web server on Linux, you can configure and start Apache from YaST.

- 1 In YaST, click *Network Services* in the left pane, then click *HTTP Server* in the right pane.

Apache is referred to as the HTTP Server on Linux.



- 2 Configure the following HTTP Server settings:

- ♦ Port
- ♦ Network devices
- ♦ Modules for scripting (PHPS, Perl, Python, Ruby)
- ♦ Default host
- ♦ Virtual host

For instructions, see “Configuring Apache with YaST” (http://www.suse.com/documentation/sles11/book_sle_admin/data/sec_apache2_configuration.html#sec_apache2_configuration_yast) in the *SLES 11 Administration Guide* (http://www.suse.com/documentation/sles11/book_sle_admin/data/book_sle_admin_pre.html)

- 3 To verify that Apache has started successfully, open a Web browser and view the following URL:

`http://Web_server_address:port_number`

where *Web_server_address* is the IP address or hostname of your Linux server and *port_number* is the Apache listen port, which is 80 by default. For example:

`http://192.168.1.18:80`

`http://localhost:80`

If Apache is correctly set up, you should see a Web page that starts with the following message:

If you can see this, it means that the installation of the Apache Web Server software on this system was successful. You may now add content to this directory and replace this page.

4 Secure your Web solution by requiring strong ciphers.

- 4a** In a text editor, modify the `/etc/apache2/vhosts.d/vhost-ssl.conf` file to require strong settings by placing a plus sign (+) before RSA, HIGH, and SSLv2 and exclamation marks (!) before the weaker ciphers as follows:

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

- 4b** Restart Apache.

5 Replace the Apache test page (`/apache2/htdocs/index.html`) with your own home page and begin building your Web site.

For more information about Apache on Linux, see “Apache HTTP Server” (http://www.suse.com/documentation/sles11/book_sle_admin/data/cha_apache2.html) in the *SLES 11 Administration Guide* (http://www.suse.com/documentation/sles11/book_sle_admin/data/book_sle_admin_pre.html).

3.4 Additional Information

After you installed the Web Services and Applications packages of Linux, see their individual documentation sites for more detailed information:

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

Configuring MySQL with Novell Cluster Services

4

Novell Open Enterprise Server 11 provides an open source version of MySQL 5.0.x software that is offered under the GNU General Public License (GPL) Version 2. 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*, the open source MySQL packages require additional support contracts to be obtained by the customer in order to receive full support.

Novell also offers the MySQL Network product that combines the latest MySQL Pro Certified Server software with a comprehensive set of services and support from Novell. For information, see [MySQL Network \(http://www.novell.com/products/mysql/\)](http://www.novell.com/products/mysql/).

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 5.0 documentation for information about configuring, managing, and using MySQL. For information, see [MySQL Documentation Library: MySQL Reference Manuals \(http://dev.mysql.com/doc/\)](http://dev.mysql.com/doc/).

The instructions in this section describes how to set up MySQL on an OES 11 cluster.

- ♦ [Section 4.1, “Prerequisites for Clustering MySQL,” on page 30](#)
- ♦ [Section 4.2, “Installing and Enabling MySQL,” on page 30](#)
- ♦ [Section 4.3, “Creating an LVM Volume Group and Logical Volume,” on page 33](#)
- ♦ [Section 4.4, “Configuring MySQL on the LVM Logical Volume,” on page 38](#)
- ♦ [Section 4.5, “Cluster-Enabling MySQL on the Logical Volume,” on page 39](#)
- ♦ [Section 4.6, “File Location,” on page 42](#)
- ♦ [Section 4.7, “Security Considerations for the MySQL Configuration,” on page 43](#)
- ♦ [Section 4.8, “Additional Information,” on page 44](#)

4.1 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 and Configuring Novell Cluster Services on OES 11](#)” in the *OES 11: Novell Cluster Services 2.0 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 4.2, “Installing and Enabling MySQL,”](#) on page 30.
- ❑ 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 4.3, “Creating an LVM Volume Group and Logical Volume,”](#) on page 33.

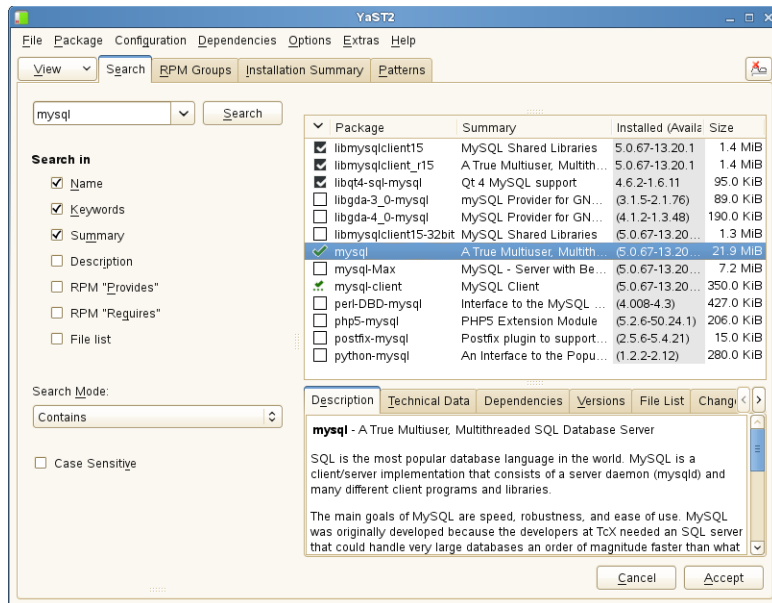
4.2 Installing and Enabling MySQL

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	<p>Provides the MySQL software, database, and the following features for users that require transaction support:</p> <ul style="list-style-type: none">♦ Berkeley database (BDB) tables♦ InnoDB tables <p>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.</p>
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 Make sure that the SUSE Linux Enterprise Server 11 SP1 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 4.3, "Creating an LVM Volume Group and Logical Volume,"](#) on page 33.
- ♦ Creates the default MySQL configuration file (`/etc/my.cnf`).

- 7 In YaST, enable the MySQL service:

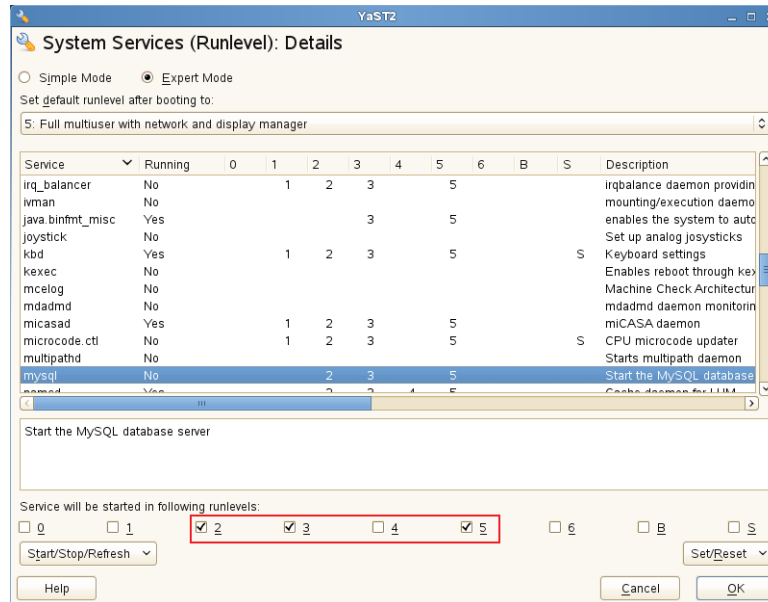
7a Select *System > System Services (Runlevel)*.

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 4.3, “Creating an LVM Volume Group and Logical Volume,”](#) on page 33.

4.3 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 4.3.1, “Sample Values,” on page 33](#)
- ♦ [Section 4.3.2, “Setting Up the VG and LV \(Overview\),” on page 33](#)
- ♦ [Section 4.3.3, “Setting up the VG and LV \(Detailed\),” on page 34](#)

4.3.1 Sample Values

The procedures in this section uses the following parameters. Make sure to 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	mysqllv
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

4.3.2 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. Make sure to substitute your own values in the commands. For details, see [Section 4.3.3, “Setting up the VG and LV \(Detailed\),” on page 34](#).

Command Action	Command
1. Create the LVM physical volume.	<code>pvcreate /dev/sdd</code>
2. Create the clustered LVM volume group.	<code>vgcreate -c y mysqlvg /dev/sdd</code>

Command Action	Command
3. Activate the volume group exclusively on the node.	<code>vgchange -a ey mysqlvg</code>
4. Create the LVM logical volume.	<code>lvcreate -n mysql1v -L size mysqlvg</code>
5. Add a file system to the LVM logical volume.	<code>mkfs -t ext3 /dev/mysqlvg/mysql1v [fs_options]</code>
6. Create a mount point for the logical volume.	<code>mkdir /mnt/mysql</code> You must also create this path on each node in the cluster.
7. Mount the LVM logical volume.	<code>mount -t ext3 /dev/mysqlvg/mysql1v / mnt/mysql</code>
8. Create the directory structure for the MySQL database files on the mounted logical volume.	<code>cd /mnt/mysql mkdir /mnt/mysql/var mkdir /mnt/mysql/var/lib mkdir /mnt/mysql/var/lib/mysql</code>
9. Modify the file ownership of the mount point and subdirectories.	<code>chown -R mysql:mysql /mnt/mysql</code>
10. Deactivate the LVM logical volume.	<code>vgchange -a n mysqlvg</code>

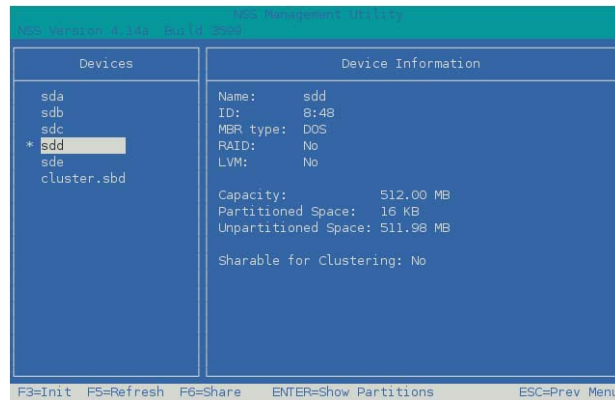
4.3.3 Setting up the VG and LV (Detailed)

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-l007-jBH3-qHvySe
```

**7** Create an LVM logical volume (such as `mysql1v`) 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 "mysql1v" -L 500 "mysqlvg"
Logical volume "mysql1v" created
```

This volume's full path name is `/dev/mysqlvg/mysql1v`.

**8** View information about the logical volume by entering:

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

For example:

```
lvdisplay -v /dev/mysqlvg/mysql1v
Using logical volume(s) on command line
--- Logical volume ---
LV Name /dev/mysqlvg/mysql1v
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/mysql1v
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/mysql1v /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 4.4, “Configuring MySQL on the LVM Logical Volume,”](#) on page 38.

## 4.4 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 [Step 6 on page 39](#) of the following procedure. Afterwards, make sure to modify the ownership of the folder and files to the `mysql` user and group by using the `chown` command as illustrated in [Step 13 of Section 4.3, "Creating an LVM Volume Group and Logical Volume," on page 33](#).

---

**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 [Section 4.5, “Cluster-Enabling MySQL on the Logical Volume,”](#) on page 39.

## 4.5 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. Make sure to replace the sample values with your values.

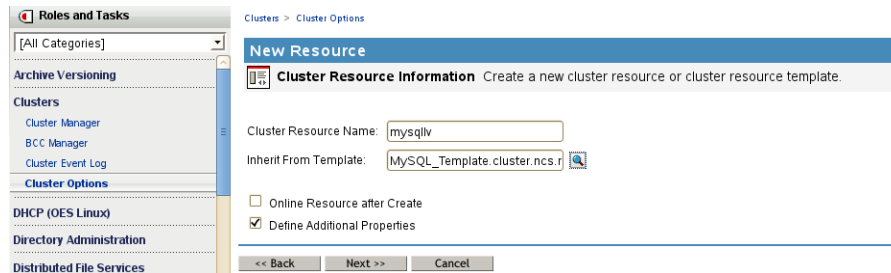
| Parameter           | Sample Value                                                                                                   |
|---------------------|----------------------------------------------------------------------------------------------------------------|
| Resource IP Address | 10.10.10.44                                                                                                    |
| MOUNT_FS            | ext3<br><br>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/mysql1v                                                                                   |
| 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 > Cluster Options*, then browse to select the cluster.
- 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/ncsfuns

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/mysql1v
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/mysql1v
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/mysql1v
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 11: Novell Cluster Services 2.0 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 *Clusters > Cluster Manager*, select the MySQL resource check box, then click *Online*.

|                                                                                     |                                                                                     |                         |                                                                                     |         |        |   |                        |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------|-------------------------------------------------------------------------------------|---------|--------|---|------------------------|
|  |  | <a href="#">mysqliv</a> |  | Running | avalon | 2 | Jul 8, 2011 5:32:48 PM |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------|-------------------------------------------------------------------------------------|---------|--------|---|------------------------|

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

## 4.6 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<br>/etc/mysqlaccess.conf                                                                               |
| Database files                                     | /var/lib/mysql                                                                                                     |
| Man pages                                          | /usr/share/man/man1                                                                                                |
| Documentation ( <i>MySQL Readme</i> )              | /usr/share/doc/packages/mysql                                                                                      |
| Log file                                           | /var/lib/mysql/mysqld.log<br><br>The MySQL log file can also be accessed via a hard link from /var/log/mysqld.log. |

| MySQL Component | Default Location in OES                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Software        | <p>Some of the software components might not appear in this location until after you enable the service.</p> <pre> /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 </pre> |

## 4.7 Security Considerations for the MySQL Configuration

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

- [Section 4.7.1, “MySQL Ports,” on page 43](#)
- [Section 4.7.2, “Securing MySQL,” on page 43](#)

### 4.7.1 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.

### 4.7.2 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.

## 4.8 Additional Information

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

- ♦ *MySQL 5.0 Reference Manual* (<http://dev.mysql.com/doc/refman/5.0/en/>) from the [MySQL Documentation Library](http://dev.mysql.com/doc/) (<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_privilege_tables(1)`
- ♦ `mysqlimport(1)`

- ♦ `mysql_install_db(1)`
- ♦ `mysqlsecure_installation(1)`
- ♦ `mysqlshow(1)`
- ♦ `mysqlupgrade(1)`

