

Web and Application Services Overview

Open Enterprise Server 2 SP3

June 21, 2012

Novell.

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 © 2005–2012 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.

Contents

About This Guide	5
1 OES Web and Application Services Overview	7
1.1 Introduction to Web and Application Services	7
1.1.1 What Are Web Services?	7
1.1.2 What Are Web Applications?	8
1.1.3 Web Application Tools (Java and J2EE)	9
1.1.4 Enabling Technologies	9
1.1.5 General Web and Application Services Architecture	11
1.2 OES Components That Provide Web and Application Services	12
1.2.1 Web Hosting: Apache Web Server 2.0	13
1.2.2 Servlet Support: Tomcat Servlet Container	14
1.2.3 Scripting: PHP and Perl	14
1.2.4 Web Databases: MySQL	14
1.2.5 Custom Web/J2EE Application: JBoss	15
1.2.6 Web and Network Search Capability: QuickFinder Server	15
1.3 What's Next	16
2 Linux: Deploying Web and Application Services Components	17
2.1 Setting Up a Simple Web Server on Linux	17
2.2 Installing Web and Application Services Packages in YaST	19
2.3 Managing Your Web Server on Linux	20
2.4 Additional Information	21
A Documentation Updates	23
A.1 June 21, 2012	23
A.2 November 21, 2011	23
A.3 December 2010 (OES 2 SP3)	23
A.4 November 9, 2009 (OES 2 SP2)	23
A.5 December 2008 (OES 2 SP1)	24
A.6 December 1, 2005 (OES 2)	24

About This Guide

Novell Open Enterprise Server (OES) 2 includes a collection of open source and Novell 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.

The guide is divided into the following sections:

- ♦ [Chapter 1, “OES Web and Application Services Overview,” on page 7](#)
- ♦ [Chapter 2, “Linux: Deploying Web and Application Services Components,” on page 17](#)
- ♦ [Appendix A, “Documentation Updates,” on page 23](#)

Audience

This guide introduces you to Web and application services, and explains how you can begin using them to meet the demands of your business. It is intended for Web or network administrators who install and manage Web site content and applications. Developers who write Web-based applications to run in the OES environment might also find the information in this overview helpful.

Feedback

We want to hear your comments and suggestions about this guide and the other documentation included with Novell OES. Please use the User Comment feature at the bottom of each page of the OES online documentation.

Documentation Updates

For the most recent documentation, visit the [OES 2 Documentation Web site \(http://www.novell.com/documentation/oes2\)](http://www.novell.com/documentation/oes2).

Additional Documentation

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 4 documentation \(http://jakarta.apache.org/tomcat/tomcat-4.1-doc\)](http://jakarta.apache.org/tomcat/tomcat-4.1-doc)
- ♦ [MySQL documentation \(http://dev.mysql.com/doc\)](http://dev.mysql.com/doc)

For information about Novell exteNd, see the [Novell exteNd documentation Web site \(http://www.novell.com/documentation-index/index.jsp?category=exteNd\)](http://www.novell.com/documentation-index/index.jsp?category=exteNd).

1 OES Web and Application Services Overview

Novell Open Enterprise Server (OES) 2 includes a collection of open source and Novell 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. Using OES, you can use the full range of Web and application services.

This section covers the following topics:

- [Section 1.1, “Introduction to Web and Application Services,” on page 7](#)
- [Section 1.2, “OES Components That Provide Web and Application Services,” on page 12](#)
- [Section 1.3, “What’s Next,” on page 16](#)

1.1 Introduction to Web and Application Services

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 services and Web applications.

- [Section 1.1.1, “What Are Web Services?,” on page 7](#)
- [Section 1.1.2, “What Are Web Applications?,” on page 8](#)
- [Section 1.1.3, “Web Application Tools \(Java and J2EE\),” on page 9](#)
- [Section 1.1.4, “Enabling Technologies,” on page 9](#)
- [Section 1.1.5, “General Web and Application Services Architecture,” on page 11](#)

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 10, 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 9](#)
- ♦ [“Servlet Containers” on page 10](#)
- ♦ [“Web Database Servers” on page 10](#)
- ♦ [“Application Servers” on page 10](#)

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

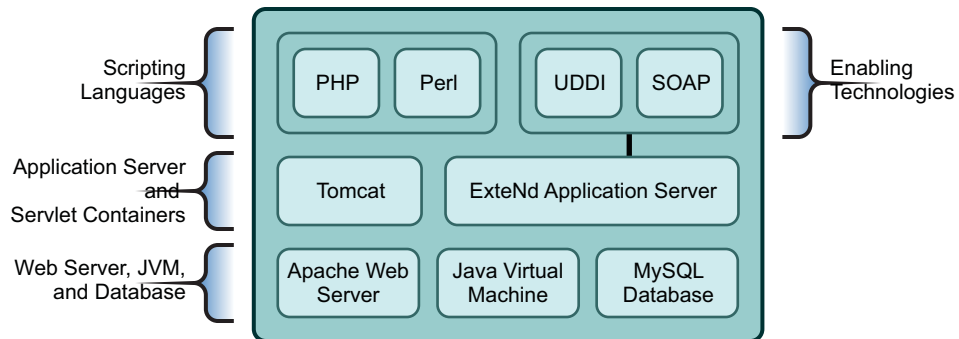
In more sophisticated Web application models, an application server is added to enable the system to manage business logic and track the user's progress through the application. The application server software runs in a middle tier, between Web browser-based clients and back-end databases and business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

Examples of J2EE application servers are the open source JBoss application server and the commercial Novell exteNd Application Server.

1.1.5 General Web and Application Services 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*

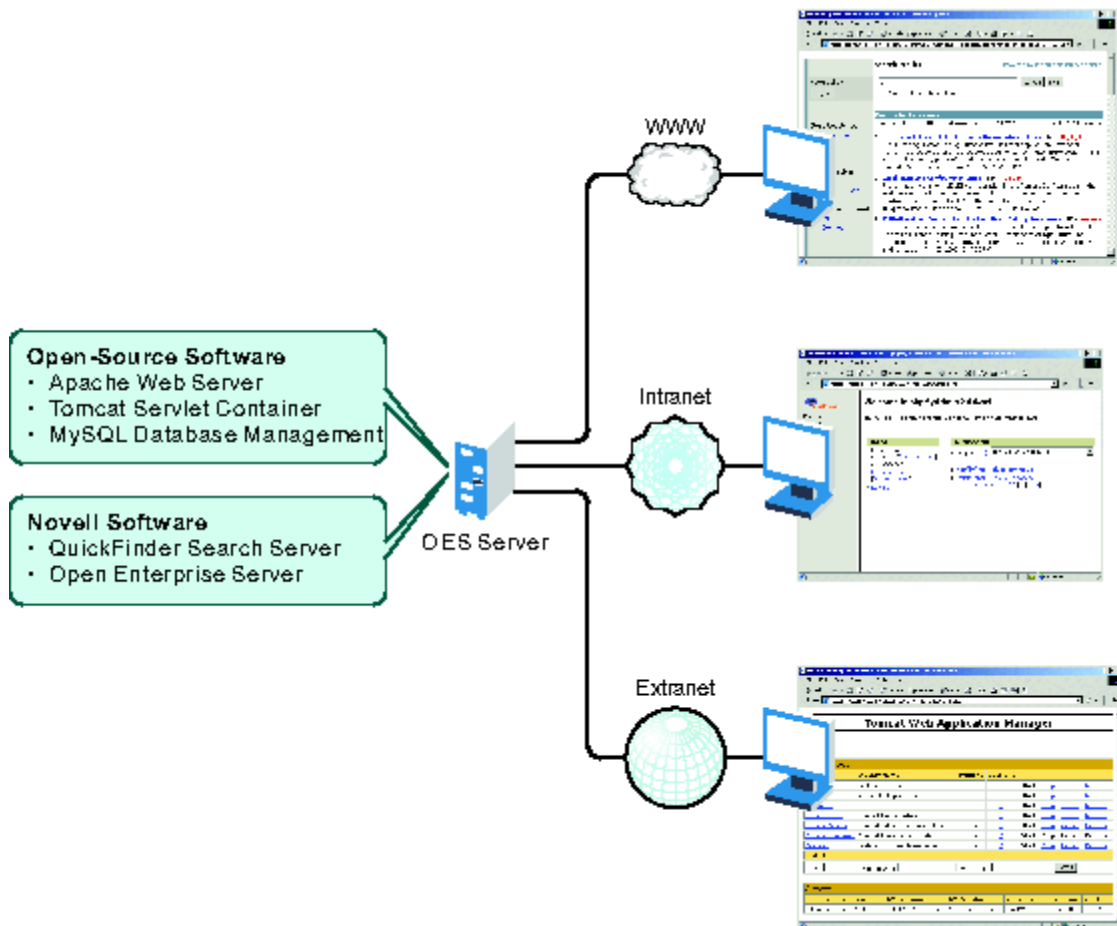


1.2 OES Components That Provide Web and Application Services

OES comes bundled with all of the Web and application services 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 or Novell exteNd 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 and application services component included with OES:

- ♦ [Section 1.2.1, “Web Hosting: Apache Web Server 2.0,” on page 13](#)
- ♦ [Section 1.2.2, “Servlet Support: Tomcat Servlet Container,” on page 14](#)
- ♦ [Section 1.2.3, “Scripting: PHP and Perl,” on page 14](#)
- ♦ [Section 1.2.4, “Web Databases: MySQL,” on page 14](#)
- ♦ [Section 1.2.5, “Custom Web/J2EE Application: JBoss,” on page 15](#)
- ♦ [Section 1.2.6, “Web and Network Search Capability: QuickFinder Server,” on page 15](#)

1.2.1 Web Hosting: Apache Web Server 2.0

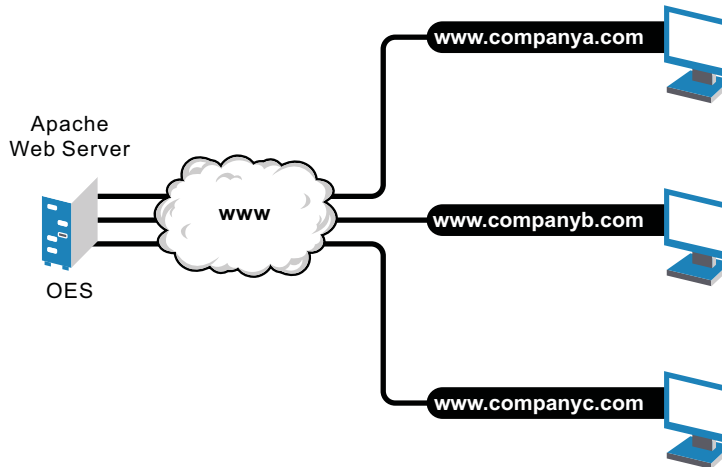
Apache is the most popular Web server being used on the World Wide Web today. Its popularity comes from the fact that it is the most reliable and secure Web server available. It runs on all major platforms, is capable of hosting even the most complex Web sites, and can scale to handle thousands of simultaneous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

The Apache Web Server 2.0 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 2.0 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 5 for Linux, which implements the Java Servlet 2.4 and JSP 2.0 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.

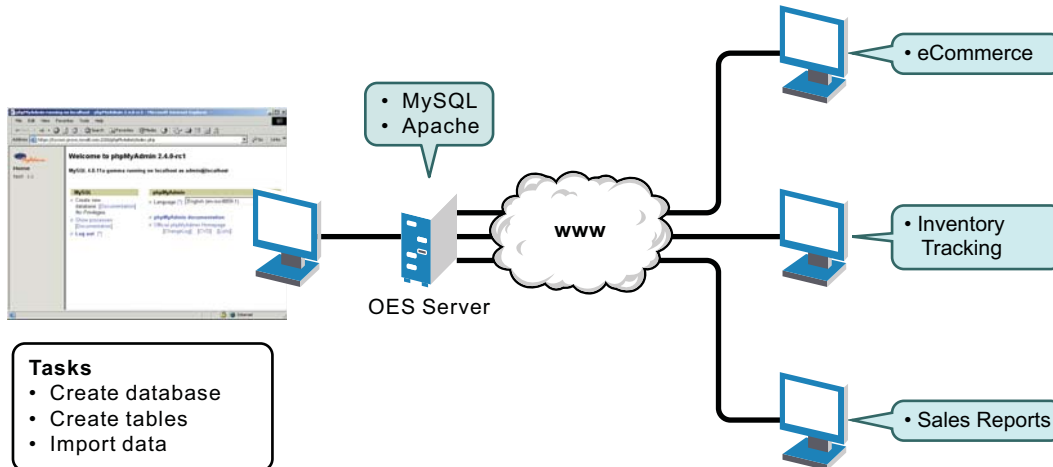
1.2.4 Web Databases: MySQL

OES includes the open source MySQL 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 Custom Web/J2EE Application: JBoss

When you need greater processing power beyond what scripting or Web application hosting with Tomcat can offer, OES offers a J2EE-certified application servers: JBoss. Bundled with SLES 10, JBoss provides enterprise-class security, transaction support, resource management, load balancing, and clustering. JBoss application server is a comprehensive, J2EE-certified platform for building and deploying enterprise-class Web applications. It supports JSP, EJBs, and all other standard J2EE components and technologies.

1.2.6 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.

1 User enters search query on www.digitalairlines.com.

2 Search is sent to search.digitalairlines.com which is hosted by QuickFinder Server.

3 QuickFinder Server sends search results list to user.

4 User clicks a search result to retrieve document from www.digitalairlines.com.

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 2 SP3: Installation Guide*.

2 Linux: Deploying Web and Application Services Components

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

- ♦ [Section 2.1, “Setting Up a Simple Web Server on Linux,” on page 17](#)
- ♦ [Section 2.2, “Installing Web and Application Services Packages in YaST,” on page 19](#)
- ♦ [Section 2.3, “Managing Your Web Server on Linux,” on page 20](#)
- ♦ [Section 2.4, “Additional Information,” on page 21](#)

2.1 Setting Up a Simple Web Server on Linux

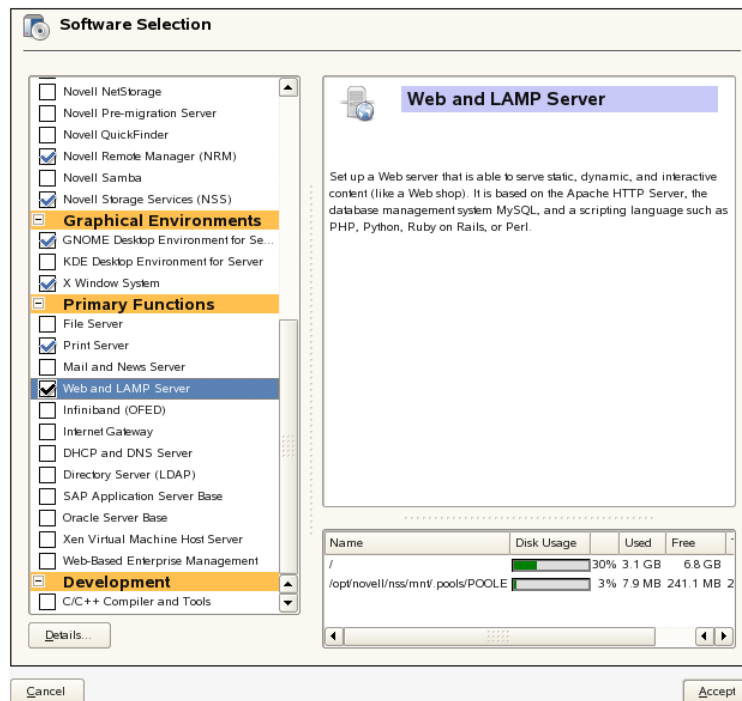
You can install the Web server software during the install or afterwards by selecting the Web and Lamp Server option under the Linux Primary Functions category. The Web and LAMP Server pattern includes the Apache and MySQL packages, and scripting languages such as PHP, Python, Ruby on Rails, and Perl.

For complete instructions on installing Linux services on OES 2, see [OES 2 SP3: Installation Guide](#). For instructions on configuring Linux services, see the [SLES 10 SP4 Installation and Administration Guide](#) (http://www.novell.com/documentation/sles10/book_sle_reference/data/book_sle_reference.html).

To install the Web and LAMP Server pattern on an existing OES 2 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*.



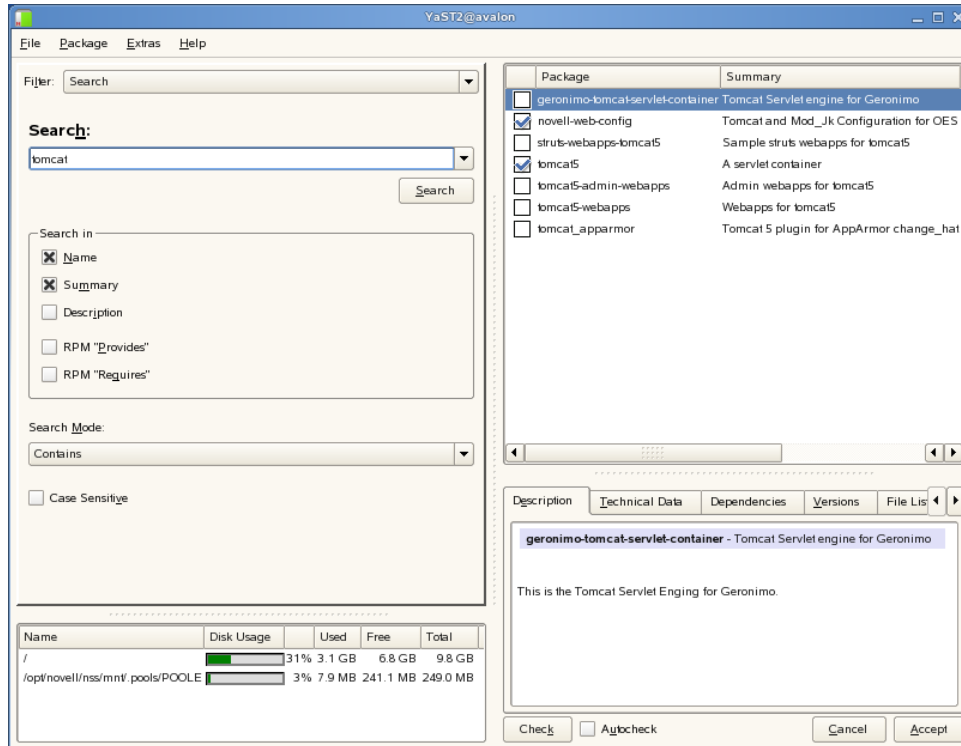
- 4 Continue with the installation.
- 5 After the install, configure the Apache Web Server.

For information, see “Apache HTTP Server” (http://www.novell.com/documentation/sles10/book_sle_reference/data/cha_apache2.html) in the *SLES 10 SP4 Installation and Administration Guide* (http://www.novell.com/documentation/sles10/book_sle_reference/data/book_sle_reference.html).

2.2 Installing Web and Application Services Packages in YaST

After you have installed Linux, you can use YaST to add more Web and application services 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.

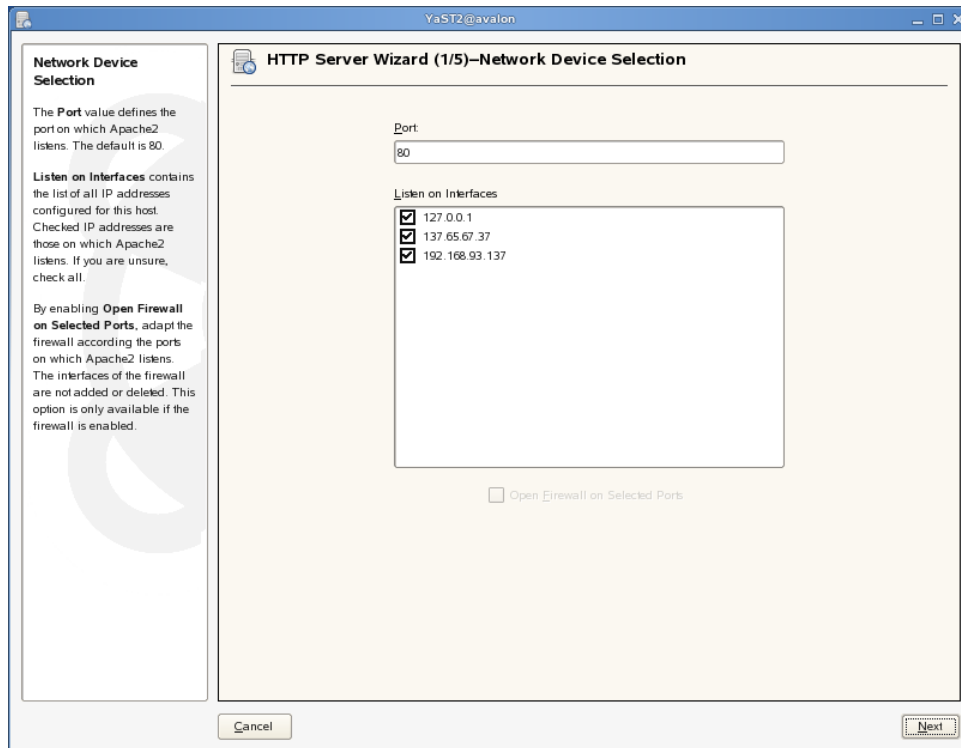
Use this process to add Web and application services packages to your Linux server as needed.

For more information about YaST, see YaST Configuration in Installation in the *SLES 10 Installation and Administration Guide* on the [SUSE LINUX Enterprise Server 10 documentation page \(http://www.novell.com/documentation/sles10\)](http://www.novell.com/documentation/sles10).

2.3 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 “40.2.2 Configuring Apache with YaST” (http://www.novell.com/documentation/sles10/book_sle_reference/data/sec_apache2_configuration.html#sec_apache2_configuration_yast)

- 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 The Apache Web Server in Services in the *SLES 10 Installation and Administration Guide* on the [SUSE LINUX Enterprise Server documentation page](http://www.novell.com/documentation/sles10) (<http://www.novell.com/documentation/sles10>).

2.4 Additional Information

After you installed the Web and application services packages of Linux, refer to their individual documentation sites for more detailed information.

- ♦ [Apache 2.0 Documentation](http://httpd.apache.org/docs/2.2/) (<http://httpd.apache.org/docs/2.2/>)
- ♦ [Tomcat 5 documentation](http://tomcat.apache.org/tomcat-5.5-doc/index.html) (<http://tomcat.apache.org/tomcat-5.5-doc/index.html>)
- ♦ [MySQL documentation](http://dev.mysql.com/doc) (<http://dev.mysql.com/doc>)
- ♦ [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>)

A Documentation Updates

This section contains information about documentation content changes made since the initial release of Novell Open Enterprise Server. If you are an existing user, review the change entries to readily identify modified content. If you are a new user, simply read the guide in its current state.

This document was updated on the following dates:

- ♦ [Section A.1, “June 21, 2012,” on page 23](#)
- ♦ [Section A.2, “November 21, 2011,” on page 23](#)
- ♦ [Section A.3, “December 2010 \(OES 2 SP3\),” on page 23](#)
- ♦ [Section A.4, “November 9, 2009 \(OES 2 SP2\),” on page 23](#)
- ♦ [Section A.5, “December 2008 \(OES 2 SP1\),” on page 24](#)
- ♦ [Section A.6, “December 1, 2005 \(OES 2\),” on page 24](#)

A.1 June 21, 2012

We recommend that you use strong ciphers for your Apache Web Server. See [Step 4](#) in [Section 2.3, “Managing Your Web Server on Linux,” on page 20](#).

A.2 November 21, 2011

In addition to bug fixes, Novell Cluster Services added support for OES 2 SP3 services and file systems on the SUSE Linux Enterprise Server (SLES) 10 SP4 operating system. You can upgrade to SLES 10 SP4 by using the move-to-sles10-sp4 patch in the SLES patch channel.

Links have been altered to the SLES 10 SP4 documentation Web site.

A.3 December 2010 (OES 2 SP3)

This guide was updated to conform with Novell documentation standards. Information specific to the NetWare 6.5 SP8 operating system was removed. For NetWare Web information, see the [NW 6.5 SP8: Web and Application Services Overview](#).

A.4 November 9, 2009 (OES 2 SP2)

Updated to the revised documentation standards.

A.5 December 2008 (OES 2 SP1)

Updated to the revised documentation standards

A.6 December 1, 2005 (OES 2)

Page design reformatted to comply with revised Novell documentation standards.