

Pivotal™ Web Server®

Version 6.0

Installation and Configuration

Rev: 02

© 2014 Pivotal Software, Inc.

Notice

Copyright

Copyright © 2016 Pivotal Software, Inc. All rights reserved.

Pivotal Software, Inc. believes the information in this publication is accurate as of its publication date. The information is subject to change without notice. THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." PIVOTAL SOFTWARE, INC. ("Pivotal") MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any Pivotal software described in this publication requires an applicable software license.

All trademarks used herein are the property of Pivotal or their respective owners.

Contents

Preface: Contacting Pivotal.....	iv
Current Pivotal Customers.....	v
Public Inquiries About Application Fabric Products.....	vi
Chapter 1: About Pivotal Web Server Installation and Configuration....	7
Intended Audience.....	8
Overview of Pivotal Web Server.....	9
Information for Internationalized Applications (Non US-ASCII Deployments).....	9
Complete Packages and Modules in Pivotal Web Server 5.4.2 6.0.1.....	9
Differences Between Pivotal Web Server and Pivotal ERS.....	10
Installing Pivotal Web Server.....	12
Available Distribution Packages.....	12
RHEL: Install Pivotal Web Server from an RPM.....	12
Unix: Install Pivotal Web Server from a Self-Extracting ZIP.....	14
Windows: Install Pivotal Web Server from a ZIP File.....	15
Windows: Install Pivotal Web Server Using the MSI Installer.....	16
Description of the Pivotal Web Server Installation.....	18
Upgrading Pivotal Web Server.....	20
Upgrade Pivotal Web Server 5.4.x.....	20
Migrating from VMware vFabric Web Server 5.x.....	21
Retain the Current Installation Path and Instance Directories.....	21
Update the Installation Path, but Retain the Current Instance Directories.....	22
Update the Installation Path and Relocate All Instances.....	23
Migrating Enterprise Ready Server to Pivotal Web Server.....	24
Preparing to Migrate.....	24
Migration Procedure.....	29
Creating and Using Pivotal Web Server Instances.....	31
Description of Pivotal Web Server Instances.....	31
Create Pivotal Web Server Instances.....	31
newsrver Prompts and Command Reference.....	33
Unix: Start and Stop Pivotal Web Server Instances.....	38
Windows: Start and Stop Pivotal Web Server Instances.....	40
httpdctl Reference.....	41
Serve a Sample HTML File from Your Pivotal Web Server Instance.....	42
Configuring Pivotal Web Server Instances.....	44
Using Sample Configuration Files to Enable Features and Modify Configuration.....	44
Configure Load Balancing Between Two or More tc Runtime Instances.....	45
Configure SSL Between Pivotal Web Server and Pivotal tc Server.....	48
Configure BMX for Monitoring Pivotal Web Server Instances.....	52
Metrics.....	53
Security Information.....	57
External Interfaces, Ports, and Services.....	57
Resources That Must Be Protected.....	57
Log File Locations.....	57
User Accounts Created at Installation.....	58
Obtaining and Installing Security Updates.....	58
Additional Documentation.....	59

Contacting Pivotal

Current Pivotal Customers

Submit a ticket from the [Help & Support Page](#).

Public Inquiries About Application Fabric Products

Email the appropriate group for the Application Fabric product:

- RabbitMQ@goPivotal.com
- tcServer@goPivotal.com
- WebServer@goPivotal.com

Chapter 1

About Pivotal Web Server Installation and Configuration

Pivotal Web Server Installation and Configuration describes product concepts and product-specific configuration tasks for Pivotal Web Server. Fully compatible with Apache Web Server, Pivotal Web Server is a dynamic load-balancing service that is available as a standalone product. Read this documentation for an overview of Pivotal Web Server features, installation instructions, and information on how to configure functionality that is specific to Pivotal Web Server.

Intended Audience

Pivotal Web Server Installation and Configuration is intended for experienced Windows and Unix developers and system administrators who want to install a Web Server to serve static Web content, act as a proxy, or load balance between application servers such as Pivotal tc Server.

Overview of Pivotal Web Server

Pivotal Web Server is based on Apache HTTP Server, providing a familiar, proven foundation for your web tier. In addition to the standard features of Apache HTTP Server, Pivotal Web Server provides the following benefits:

- Ability to easily install multiple instances of Pivotal Web Server running on a single computer.
- Scalable management of multiple Web sites and servers; you can run and manage hundreds of instances of Pivotal Web Server.
- Support for heterogeneous environments (32- and 64-bit architectures): Red Hat Enterprise Linux (RHEL), Solaris, Ubuntu, and Microsoft Windows. Pivotal Web Server can be updated and patched across all servers at once.

Subtopics

Information for International Users

Complete Packages and Modules in Pivotal Web Server 5.5 6.2

Differences Between Pivotal Web Server and Pivotal ERS

Information for Internationalized Applications (Non US-ASCII Deployments)

If you require multi-byte character-set support, Pivotal recommends that you follow these guidelines for using multi-byte filenames and multi-byte characters in configuration files. These guidelines provide the most flexibility and most consistent results:

- Use the UTF-8 encoding when running a Unix terminal or Windows PowerShell. For example, to set this encoding on Unix, you can use the following environment variable:

```
LANG=en_US.UTF-8
```

On Windows, you can run the following command in your PowerShell window:

```
PS prompt> chcp 65001
```

Important: The preceding Windows command does not always work as expected on pre-R2 versions of Windows 2008. For this reason, Pivotal strongly recommends that you install Pivotal Web Server on Windows 2008 R2.

Setting your encoding to UTF-8 results in better display of `httpctl` output, easier examination of the log files, and so on.

- Edit and save all Pivotal Web Server configuration files, such as `conf/httpd.conf`, in UTF-8 format. This improves the legibility of the Web Server access and error log files.

Complete Packages and Modules in Pivotal Web Server 5.4.2 6.0.1

This section lists the complete contents of Pivotal Web Server 5.4.2 6.0.1 .

- Apache Web Server httpd 2.2.29 2.4.10
- Apache mod_ftp 0.9.6
- Apache mod_fcgid 2.3.7 2.3.9
- Apache mod_jk 1.2.40
- Apache APR Library 1.5.1 1.5.1
- Apache APR-util Library 1.5.3

- Apache tcnative connector
- Expat 2.1.0
- GNU libiconv 1.11
- Libxml2 2.6.32.1
- Lua 5.2.2
- mod_bmx 0.9.4 (Hyperic plug-in for monitoring support)
- OpenSSL 1.0.1j 1.0.1j
- OpenSSL/FIPS 2.0
- OpenLDAP 2.3.43
- PCRE 8.32
- zlib 1.2.7

You typically install Pivotal Web Server from the `pivotal-web-server-version-platform` package, which contains all the preceding compiled modules. For your convenience, Pivotal also makes a `pivotal-web-server-devel-version-platform` package available which you can use to build HTTPD modules. Typically, you install the `-devel` package only on development computers, and not on production computers.

mod_fcgid Implementation of Connector to FastCGI

The `mod_fcgid` distributed with Pivotal Web Server is an implementation of the connector to FastCGI applications. This module allows the user to provision FastCGI providers such as PHP or Ruby on Rails from third parties, running out-of-process from the server itself.

Many applications can be built to support FastCGI; consult your language or application documentation for details. The application providing FastCGI services is launched by `mod_fcgid` on the initial request, and reused for subsequent requests to that application or language environment.

For details on configuring an application, including the number of persistent processes created, see *Apache Module mod_fcgid*.

Differences Between Pivotal Web Server and Pivotal ERS

The Pivotal Cloud Application Platform includes two HTTP server and load-balancing products: Pivotal Web Server and Pivotal Enterprise Ready Server (ERS).

Important: The end of general support (EOGS) date for Pivotal ERS is July 1, 2014. Pivotal highly recommends that ERS customers migrate to Pivotal Web Server. The following table describes the major differences between the two products and provides high-level actions that existing ERS customers can take as they prepare for the migration.

Table 1: Differences Between Pivotal Web Server and Pivotal ERS

Pivotal Web Server	Pivotal ERS	Customer Action
Available as a standalone product.	Originally designed for only physical computers.	Select licensing based on Pivotal integration or dedicated hardware.
Strictly an Apache HTTPD Server-based product. Pivotal tc Server, a separate product, is strictly an Apache Tomcat-based product.	Includes both Apache HTTPD and Apache Tomcat packages.	Migrate ERS Apache HTTPD instances to Pivotal Web Server. Separately migrate ERS Tomcat instances to Pivotal tc Server.

Pivotal Web Server	Pivotal ERS	Customer Action
Runs on current, vendor-supported 32- and 64-bit releases of RHEL, Microsoft Windows, Solaris, and AIX. See <i>Supported Configurations and System Requirements</i> for the exact versions.	Runs on now-unsupported, or "twilighted", versions of RHEL, Windows, Solaris, AIX, and HPUX. See <i>ERS Supported Platforms</i> for the exact versions.	Upgrade to a vendor-supported operating system version for all Pivotal Web Server instances, and apply all patch releases (such as service packs) issued by that vendor no later than 12 months from their vendor release.
Includes the current enterprise-ready release of Apache HTTPD Server 2.2.2.4 .	Includes the current release of Apache HTTPD Server 2.2.2.4 , as well as the now-deprecated 2.0 and 1.3 versions.	Migrate all Apache HTTPD 2.0 and 1.3 instances to Pivotal Web Server 2.2.2.4 instances. The migration requires updates to the *.conf file.
During installation or upgrade, the Apache HTTPD binaries are written to a path in the format <code>pivotal-web-server/httpd-2.2.2.4.xx.x-32</code> . This preserves any existing Apache HTTPD binaries without overwriting them.	During installation or upgrade, the Apache HTTPD binaries are always written to the same directory (<code>ers-install-path/apache-2.2.2.4-64</code>), which means on upgrade any existing binaries are overwritten.	Point all server instances to the common symlink <code>pivotal-web-server/httpd-2.2.2.4</code> , modify it to revert/roll back/change 32-64 bit modes.
Includes the most commonly-used modules. See <i>Complete Packages and Modules in Pivotal Web Server 5.5.6.2</i> .	Included additional modules, such as <code>mod_perl</code> , <code>mod_php</code> , and <code>mod_snmp</code> .	Migrate PHP and Perl applications to the supported, and more optimal, <code>mod_fcgid</code> environment.
Closely tracks Apache Software Foundation (ASF) naming and directory layout conventions. In particular: <ul style="list-style-type: none"> • <code>install-dir/httpd-2.2.2.4/modules/</code> directory contains the loadable modules • Binaries and configuration file names use <code>httpd</code> prefix • <code>install-dir/newserver</code> creates new instances • <code>instance-dir/bin/httpdctl</code> controls each deployed instance • <code>instance-dir/conf/extras/</code> offers feature-based small config templates 	Has a number of now-stale, legacy file and path conventions. In particular: <ul style="list-style-type: none"> • <code>install-dir/apache-2.2.2.4/modules/standard</code> directory contains the loadable modules • Binaries and configuration file names use <code>httpsd</code> prefix • <code>install-dir/ers-server.pl</code> creates new instances • <code>install-dir/servers/instance-dir/bin/apache_startup.sh bat</code> controls each deployed instance • <code>install-dir/servers/instance-dir/conf/httpsd.conf</code> is one large, monolithic configuration template. 	Create a new Pivotal Web Server instance, then migrate customizations from your existing Pivotal ERS instance. Alternatively, modify a copy of the deployed Pivotal ERS instance tree to use Pivotal Web Server path and file name conventions. Use smaller functional *.conf snippets to make the configuration more organized and maintainable.

Installing Pivotal Web Server

Pivotal Web Server has several options for installation and setup. Installation options vary according to whether your operating system is Unix, Windows, or another supported platform.

Subtopics

Available Distribution Packages

RHEL: Install Pivotal Web Server from an RPM

Unix: Install Pivotal Web Server from a Self-Extracting ZIP

Windows: Install Pivotal Web Server from a Self-Extracting ZIP File

Windows: Install Pivotal Web Server Using the MSI Installer

Description of the Pivotal Web Server Installation

Available Distribution Packages

Pivotal Web Server is split into the following two distribution packages to simplify the installation and deployment to your datacenter:

- `pivotal-web-server`: Base package entirely sufficient for all production environments.
- `pivotal-web-server-devel`: Supplemental package that developers can use to compile and link `http` modules with the same headers and libraries as `httpd` itself using the `httpd- 2.2 2.4 /bin/apxs` tool. This package requires that you also install the base package.

RHEL: Install Pivotal Web Server from an RPM

Pivotal recommends that you install Pivotal Web Server on a Red Hat Linux Enterprise (RHEL) computer by first installing the Pivotal RPM repository and then using `yum` to perform the actual installation. See *Install Pivotal Web Server from the Pivotal RPM Repository*.

You can also download the RPM from the Pivotal download page, and install it on your RHEL computer using the `rpm` command, as described in *Install Pivotal Web Server from a Downloaded RPM*.

Install Pivotal Web Server from the Pivotal RPM Repository

Pivotal recommends that you install Web Server on RHEL computers using the Pivotal RPM repository.

Prerequisites

- Verify that your system meets the supported configurations and installation requirements. See *Supported Configurations and System Requirements*.
- Install the Pivotal repository RPM, which makes it easier for you to browse the Pivotal RPMs, including the Pivotal Web Server RPM. You install the Pivotal repository RPM on each RHEL computer on which you want to install one or more Pivotal products, such as Pivotal tc Server.
 1. On the RHEL computer, start a terminal as the `root` user.
 2. Install the Pivotal repository RPM using the following `wget` command, passing it the appropriate URL. The URL differs depending on the version of RHEL you are using.

Important: You must run the entire `wget` command on a single line. Be sure you include the `| sh` at the end, or the RPM installation will not work.

RHEL 5:

```
prompt# wget -q -O - http://packages.gopivotal.com/pub/rpm/rhel5/app-
suite/app-suite-installer | sh
```

RHEL 6:

```
prompt# wget -q -O - http://packages.gopivotal.com/pub/rpm/rhel6/app-
suite/app-suite-installer | sh
```

The command performs the following tasks:

- Imports the Pivotal GNU Privacy Guard (GPG) key.
 - Launches the Pivotal End User License Agreement (EULA) acceptance and repository configuration script.
 - Outputs the EULA for you to read; you must answer `yes` to accept the terms and continue.
3. Use the `yum search pivotal` command to view the list of Pivotal products that you can install from the Pivotal repository. For example (output truncated for clarity):

```
prompt# yum search pivotal
...
===== Matched: pivotal
=====
pivotal-rabbitmq-java-client-bin.noarch : The RabbitMQ Java Client Library
pivotal-rabbitmq-server.x86_64 : The RabbitMQ server
pivotal-tc-server-standard.noarch : Pivotal tc Server Standard
pivotal-web-server.x86_64 : Pivotal Web Server
...
```

The Pivotal Web Server RPM is called `pivotal-web-server`.

Procedure

1. From the RHEL computer on which you will install Pivotal Web Server, log in as the `root` user and start a terminal.
2. Execute the following `yum` command:

```
prompt# yum install pivotal-web-server
```

The `yum` command begins the install process, resolves dependencies, and displays the packages it will install.

The `yum` command automatically chooses the appropriate RPM package based on your architecture (32- or 64-bit).

To install the developer's package:

```
prompt# yum install pivotal-web-server-devel
```

3. Enter `y` at the prompt to begin the actual installation.

If the installation is successful, you see a `Complete!` message at the end.

What the yum install command does

The `yum install` command:

- Installs Pivotal Web Server into the `/opt/pivotal/webserver` directory.
- Sets the owner of installation directory, along with all child directories and files, to `root:root`.

What to do next

- Read *Description of the Pivotal Web Server Installation* for a brief tour of what was installed.
- Create and start using a Pivotal Web Server as described in *Creating and Using Pivotal Web Server Instances*.

Install Pivotal Web Server From a Downloaded RPM

You can install Pivotal Web Server on RHEL by downloading the RPM from the Pivotal download center and executing the `rpm` command.

Prerequisites

- Verify that your system meets the supported configurations and installation requirements. See *Supported Configurations and System Requirements*.

Procedure

1. Log in to the RHEL computer on which you will install Pivotal Web Server as the `root` user.
2. From the *Pivotal Web Server product page*, click **Downloads**.
3. Download the Web Server RPM file to a directory on your computer. The RPM file is called `pivotal-web-server-version.architecture.rpm`.
4. Start a terminal and change to the directory in which you downloaded the RPM.
5. Execute the following `rpm` command to install Web Server:

```
prompt# rpm -ivhf pivotal-web-server-version.architecture.rpm
```

6. In the previous section, see *What the yum install command does* for post-installation information, such as the installation directory. (The `yum install` command corresponds to the `rpm` command in this procedure.)

Unix: Install Pivotal Web Server from a Self-Extracting ZIP

You can install Pivotal Web Server on Unix computers with a self-extracting ZIP file that you download from the Pivotal Web site. Self-extracting zip files expand themselves, or you can use `unzip` if your platform supports it. Using `unzip` explicitly enables you to specify options in addition to what is executed by default when the ZIP self-extracts.

Prerequisites

- Verify that your system meets the supported configurations and installation requirements. See *Supported Configurations and System Requirements*.
- If you want to use `unzip` so you can specify additional options, but your platform does not support `unzip`, *obtain an unzip command*.
- If your operating system is configured to support only 64-bit operation, an external `unzip` utility is required. Do not use the `jar` utility to unpack these zip files, because the file system permissions will not be unpacked correctly.
- Be sure you have installed Perl on your computer, and that it is at least version 5.8. Pivotal recommends that you run the Perl script (described in the procedure) in multibyte character encoding such as UTF-8, especially if you are an international customer.

Procedure

1. Log in as the `root` user on to the Unix computer on which you want to install Pivotal Web Server.
2. Create the directory in which you will install Pivotal Web Server. For example:

```
prompt# mkdir /opt/pivotal
```

3. From the *Pivotal Web Server product page*, click **Downloads**.
4. Download the appropriate Pivotal Web Server self-extracting ZIP, and place it in the directory you created.

Be sure to choose the correct Unix operating system and chip architecture. For example, the file to install Pivotal Web Server on a 64-bit Linux platform is `pivotal-web-server-version-x86_64-linux-glibc2.zip.sfx`.

- From your terminal window, change to the directory in which you downloaded the ZIP file:

```
prompt# cd /opt/pivotal
```

- If necessary, change the permissions of the downloaded ZIP file to make it executable:

```
prompt# chmod +x pivotal-web-server-version-x86_64-linux-glibc2.zip.sfx
```

- Self-extract the files from the downloaded ZIP by using the file name as a command. For example:

```
prompt# ./pivotal-web-server-version-x86_64-linux-glibc2.zip.sfx
```

When it completes, the Pivotal Web Server files are located in the `webserver` subdirectory.

- Change to the `webserver` directory and run the `fixrootpath.pl` Perl script to correct the root paths. For example:

```
prompt# cd webserver
prompt# perl fixrootpath.pl
```

What to do next

- Read *Description of the Pivotal Web Server Installation* for a brief tour of what was installed.
- Create and start using a Pivotal Web Server as described in *Creating and Using Pivotal Web Server Instances*.

Windows: Install Pivotal Web Server from a ZIP File

You install Pivotal Web Server on Windows computers using a self-extracting ZIP file (`*.zip.exe`) that you download from the Pivotal Web site.

Prerequisites

- Verify that your system meets the supported configurations and installation requirements. See *Supported Configurations and System Requirements*.
- You must use Windows PowerShell 2.0 to execute the Pivotal Web Server scripts on Windows computers. Most recent Windows versions have PowerShell installed by default, but some older versions of Windows do not. To check whether your version of Windows has PowerShell installed, go to Start > All Programs > Accessories and check for `Windows PowerShell` in the list.

If Windows PowerShell 2.0 is not installed on your Windows computer, download and install it from the *Windows Management Framework (Windows PowerShell 2.0, WinRM 2.0, and BITS 4.0)* Web page on the Microsoft Support.

- If necessary, enable Windows PowerShell for script processing; by default, script processing is disabled.

- Start PowerShell from the Start Menu as an Administrator by opening Start > All Programs > Accessories > Windows PowerShell, then right-clicking on Windows PowerShell and selecting Run as Administrator. A PowerShell window starts.

- Check the current PowerShell setting by executing the following command:

```
PS prompt> Get-ExecutionPolicy
```

If the command returns `Restricted`, it means that PowerShell is not yet enabled. Enable it to allow local script processing at a minimum by executing the following command:

```
PS prompt> Set-ExecutionPolicy RemoteSigned
```

You can choose a different execution policy for your organization if you want, as well as enable PowerShell using Group and User policies. Typically, only the Administrator will be using the Pivotal Web Server scripts, so the `RemoteSigned` execution policy should be adequate in most cases.

Procedure

1. From the Windows computer on which you want to install Pivotal Web Server, log in as the Administrator user.
2. Start PowerShell from the Start Menu as an Administrator by opening Start > All Programs > Accessories > Windows PowerShell, then right-clicking on Windows PowerShell and selecting Run as Administrator. A PowerShell window starts.
3. Create the directory into which you will install Pivotal Web Server. Do not create a directory name that contains spaces. For example:

```
PS prompt> mkdir c:\opt\pivotal
```

4. From the *Pivotal Web Server product page*, click **Downloads**.
5. Download the appropriate Pivotal Web Server self-extracting ZIP, and place it in the directory you created.

Be sure to choose the correct architecture (32- or 64-bit). For example, the file to install Pivotal Web Server on a 64-bit Windows platform is `pivotal-web-server-version-x64-windows.zip.exe`.

6. Execute the downloaded `*.zip.exe` file to self-extract the files into the directory you created. You can do this, for example, by opening Window Explorer, navigating to the directory, and double-clicking on the `*.zip.exe` file.

When the extraction completes, the Pivotal Web Server files are located in the `webserver` subdirectory.

7. From your command window, change to the main Pivotal Web Server directory:

```
PS prompt> cd c:\opt\pivotal\webserver
```

8. Run the `fixrootpath.ps1` PowerShell script to correct the root paths:

```
PS prompt> .\fixrootpath.ps1
```

9. Create a symbolic link from the existing `httpd- 2.2 2.4 .version` directory to one called `httpd- 2.2 2.4`, where `version` refers to the minor version and architecture of Pivotal Web Server.

PowerShell does not include `mklink` intrinsically, so you must use the command `cmd /c mklink`. For example, to use the 64-bit edition of Pivotal Web Server, run the following command:

```
PS prompt> cmd /c mklink /d httpd-
                2.2
                2.4
                httpd-
                2.2.29.0
                2.4.10
                .0-64
```

What to do next

- Read *Description of the Pivotal Web Server Installation* for a brief tour of what was installed.
- Create and start using a Pivotal Web Server as described in *Creating and Using Pivotal Web Server Instances*.

Windows: Install Pivotal Web Server Using the MSI Installer

Prerequisites

- Verify that your system meets the supported configurations and installation requirements. See *Supported Configurations and System Requirements*.

Procedure

1. From the *Pivotal Web Server product page*, click **Downloads**.

2. Download the 32-bit (x86) or 64-bit (x64) MSI installer distribution package to a convenient directory on your computer.
 - `pivotal-web-server-version-x86-windows.msi`
 - `pivotal-web-server-version-x64-windows.msi`
3. Log in to the computer on which you are installing Web Server as the appropriate user, such as `webserver`.
4. Navigate to the directory where you downloaded the distribution package.
5. Double-click the MSI installer.

The **Welcome** page opens.

6. Click the **Next** button.

The End-User License Agreement page opens.

7. Read the text. If you agree, then accept the terms.
8. Click the **Next** button.

The setup type page opens. The page describes two setup types:

- **Sample:** All setup options are selected for you. This setup type is intended for potential customers who are evaluating the software. This option installs a Pivotal Web Server instance named *MyWebServer* to the default location with HTTP listening on port 80. Secure Sockets Layer (SSL) is not enabled. For more control over the installation choices, choose the **Custom** option instead.
 - a. Click the **Sample** button.

The setup options list page opens.
 - b. Review the installation configuration list.
 - c. Click the **Install** button when you are ready to proceed.
- **Custom:** You can select all of the setup options.

- a. Click the **Custom** button.

The **Custom Setup** page opens.

- b. Choose to accept the default installation location for the application files, or browse and select a different location on your local hard drive.
- c. Click the **Next** button.

The **Basic Server Settings** page opens.

Complete the following fields:

- Enter a server name. The server name identifies the Web Server instance locally.
 - Enter the port number for HTTP listening.
 - Enter the host name. The host name identifies the web server instance on your network or the Internet.
 - Accept the default server location, or browse to and select a new location.
- d. Click the **Next** button.

The **SSL Settings and Administrator Email** page opens.

- e. **Optional.** Enable SSL:

- Select the **Enable SSL** checkbox if you are enabling SSL.
- Enter the port for HTTPS listening. The default value is 443.
- Browse to and select a SSL certificate file location. For information about configuring SSL between Pivotal Web Server and Pivotal tc Server, see *Configure SSL Between Pivotal Web Server and Pivotal tc Server*.
- Browse to and select a key file location.

If you do not want to enable SSL, then leave the checkbox cleared.

- f. Enter the Web Server administrator email address.

- g. Click the **Next** button.
 - h. Review the installation configuration list.
 - i. Click the **Install** button when you are ready to proceed.
9. The setup wizard:
- Installs Web Server
 - Configures and starts the Microsoft Windows services.

What to do next

- Read *Description of the Pivotal Web Server Installation* for a brief tour of what was installed.
- Create and start using a Pivotal Web Server as described in *Creating and Using Pivotal Web Server Instances*.

Uninstalling Web Server

The following options are available to uninstall Web Server:

- Click the **Start** button > **All Programs** > **Pivotal** > **Uninstall Pivotal Web Server**.
- Uninstall Pivotal Web Server using the **Programs** control panel.

Description of the Pivotal Web Server Installation

The main Pivotal Web Server directory structure, although similar in many ways to the standard Apache HTTP directory layout, differs from it in a very fundamental way: Pivotal Web Server separates the runtime binaries from the configuration data.

To implement this separation, you use the `newserver` command to create a Pivotal Web Server *instance* that lives in a subdirectory of the main Pivotal Web Server home directory. The name of the instance is the name of the new subdirectory. You then configure this instance as you want, using the standard Apache `httpd` files in the `server-name/conf` directory, such as `httpd.conf`.

Note:

Perform all configuration work inside the server instance (`server-name/conf`) directory. Never modify any files under the binary directory (such as `httpd- 2.2 2.4`).

Keeping the runtime binary files apart from the files that are configured by administrators or end users makes it easier to upgrade or apply patches to the code without the risk of overwriting or corrupting user data. It also enables administrators to run multiple server instances independently.

Directly after installing Pivotal Web Server, you see the following files and directories:

- `httpd- 2.2 2.4 /` : Symbolic link to a sibling directory that actually contains the Apache 2.2 2.4 binary runtime files.
- `httpd- 2.2 2.4 .version/` : Directory that contains the actual Apache 2.2 2.4 binary runtime files. The string specifies the Apache HTTP version, such as `2.2.29.0 2.4.10` or `2.2.29.0 2.4.10 -64`.
- `licenses/` : EULA and open source license files.
- `newserver` : (Unix) Perl script for creating Pivotal Web Server instances.
- `newserver.ps1` : (Windows) PowerShell script for creating Pivotal Web Server instances.
- `fixrootpath.pl` : (Unix) Perl script for fixing root paths; run only once.
- `fixrootpath.ps1` : (Windows) PowerShell script for fixing root paths; run only once.

After you use `newserver` to create a new Pivotal Web Server instance, the command creates a new directory that contain a separately configurable instance of Pivotal Web Server. An *instance* is a complete, discrete server configuration. You can create multiple instances. You can run multiple instances at the same time if you are careful not to use the same ports in two different instances. For example, the default HTTP listening port is 80, and only one instance on any computer is allowed to communicate on port 80 at

any one time. So if you wanted to have two Pivotal Web Server instances running at the same time on the same computer, you would configure one instance to use a port other than 80.

Each instance directory contains subdirectories that contain all the data required to run a given Pivotal Web Server instance. This includes configuration data as well as all other data that is associated with that instance's configuration. For example, assume you installed Pivotal Web Server in `/opt/pivotal/webserver` and create an instance called `myserver`:

```
prompt$ cd /opt/pivotal/webserver/myserver
prompt$ ls
bin  cgi-bin  conf  ftpdocs  htdocs  logs  proxy  ssl  var
```

The `conf` directory contains the Pivotal Web Server configuration files, such as `httpd.conf`. The `bin` directory contains the startup script used to start and stop the `myserver` instance (`httpdctl`). Each of these directories is specific to the `myserver` instance. Each instance that you create has a similar set of directories.

Upgrading Pivotal Web Server

Important: If you are upgrading from VMware vFabric Web Server 5.3.x or earlier, complete the instructions in *Migrating from VMware vFabric Web Server 5.x* instead.

- *Upgrade Pivotal Web Server 5.4.x*

Upgrade Pivotal Web Server 5.4.x

Prerequisites

- Stop all running Web Server instances in the installation that you are upgrading.
 - *Unix: Start and Stop Pivotal Web Server Instances*
 - *Windows: Start and Stop Pivotal Web Server Instances*
- 1. Move your Pivotal Web Server 5.4.x installation from its current location (typically `/opt/pivotal/webserver`) to `/opt/pivotal/webserver.5.4`.
- 2. Complete the installation instructions to deploy Pivotal Web Server to your preferred location (typically `/opt/pivotal/webserver`). See *Installing Pivotal Web Server*.
- 3. Either copy or move the `/opt/pivotal/webserver.5.4/instance` tree to `/opt/pivotal/webserver/instance`, including all file permissions and ownership.
 - Copy the tree to minimize downtime of active instances.
 - If you move the tree instead, stop and uninstall `/opt/pivotal/webserver.5.4/instance` prior to moving the directory tree, as documented in *Step 7* below.
- 4. Change to the `/opt/pivotal/webserver` directory.
- 5. Update the `httpdctl` script with new features plus the revised `pivotal httpd instance` service names:

```
./newserver --server=instance --update
```
- 6. Update the module and path names and the authorization sections in the `instance/conf/httpd.conf` file to match the 2.4 version. For guidance, see the *Apache HTTP Server documentation*.
- 7. Stop and uninstall the old instance if not performed before *Step 3*. above:

```
/opt/vmware/vfabric-web-server/instance/bin/httpdctl stop  
/opt/vmware/vfabric-web-server/instance/bin/httpdctl uninstall
```
- 8. Install and start the service from the new path:

```
instance/bin/httpdctl install  
instance/bin/httpdctl start
```
- 9. Repeat steps *Step 3* through *Step 8* for each previously deployed server instance.

Migrating from VMware vFabric Web Server 5.x

Important: If you are upgrading from Pivotal Web Server 5.4.0 or later, complete the instructions in *Upgrading Pivotal Web Server* instead.

If you are migrating from VMware vFabric Web Server 5.3.x or earlier to Pivotal Web Server 5.5 6.2, you will notice that the default installation path has changed. The path, formerly `/opt/vmware/vfabric-web-server/`, now defaults to `/opt/pivotal/webserver/`. The structure of the application and instance directories is unchanged.

There are three basic options available for migrating from VMware vFabric Web Server depending on your preferred application installation path and the preferred location of your previous user-created server instances.

- *Retain the Current Installation Path and Instance Directories*
- *Update the Installation Path, but Retain the Current Instance Directories*
- *Update the Installation Path and Relocate All Instances*

Retain the Current Installation Path and Instance Directories

This option is available if you download the `.zip.sfx/exe` self-extracting distribution files. The process is complicated by the fact that the Pivotal Web Server distributions unpack into a `webserver/` directory, and is recommended as a short-term solution only.

1. Log in to the computer as the `root` user.
2. Change to the `vfabric_web_server_dir`. For example, `/opt/vmware`.
3. Unpack the `pivotal-web-server-*.zip.sfx/exe` distribution file. Unpack the `pivotal-web-server-devel-*.zip.sfx/exe` distribution file at the same time, if desired.
4. Copy the files and symlinks links from the `webserver` folder into the `vfabric-web-server` directory.

For example:

```
cp -r /opt/vmware/webserver/* /opt/vmware/vfabric-web-server
```

5. Change to the `vfabric-web-server/` directory.
6. Verify that the `httpd- 2.2 2.4` and `newserver` symlinks refers to the correct directory and file, where `httpd-version` is the directory name moved in step 4:

```
httpd- 2.2 2.4 -> httpd-version
```

```
newserver -> httpd- 2.2 2.4 /bin/newserver.pl
```

Update Deployed Instances Using the httpdctl Script

When you upgrade Pivotal Web Server, the individual instances are not updated automatically. The `instance/bin/httpdctl` script is the most frequently refreshed component. The defacto service name changes from `vfabric-web-server` to `pivotal-web-server` with the migration.

Note:

- The instructions in this sections are required only when you upgrade your instances using the `httpdctl` script.
- If `httpdctl` is not installed as a service, then complete Step 2 only.

Complete the following steps for each instance to refresh the instance control scripts and service names:

1. Uninstall the service.

```
instance/bin/httpdctl stop
```

```
instance/bin/httpdctl uninstall
```

2. Update the service with new features plus the revised `pivotal httpd instance` service names. The following command refreshes the `httpdctl` script for the instance. The command cannot be automated, because it is impossible to determine your instance locations.

```
./newserver --update --server=instance
```

3. Reinstall the service.

```
instance/bin/httpdctl install
```

```
instance/bin/httpdctl start
```

Update the Installation Path, but Retain the Current Instance Directories

This option is available for all distributions, and is viable as a longer term solution.

1. Complete the installation instructions to deploy Pivotal Web Server to your preferred location (typically `/opt/pivotal/webserver`). See *Installing Pivotal Web Server*.
2. Log in to the computer as the `root` user.
3. Change to the `vfabric_web_server_dir`. For example, `/opt/vmware`.
4. Symlink the `/opt/pivotal/webserver/httpd-version` (`2.2.29.0 2.4.10 -64`, for example) directory as `vfabric-web-server/httpd- 2.2 2.4` , replacing any previous symlink. These `httpd` revisioned directory names are unique to each specific release of either vFabric or Pivotal Web Server.

Note: You must refresh this link manually on each subsequent Pivotal Web Server update.

5. Verify that the `httpd- 2.2 2.4` and `newserver` symlinks refers to the correct directory and file, where `httpd-version` is the directory name symlinked in step 4:

```
httpd- 2.2 2.4 -> httpd-version
```

```
newserver -> httpd- 2.2 2.4 /bin/newserver.pl
```

Update Deployed Instances Using the httpdctl Script

When you upgrade Pivotal Web Server, the individual instances are not updated automatically. The `instance/bin/httpdctl` script is the most frequently refreshed component. The defacto service name changes from `vfabric-web-server` to `pivotal-web-server` with the migration.

Note:

- The instructions in this sections are required only when you upgrade your instances using the `httpdctl` script.
- If `httpdctl` is not installed as a service, then complete Step 2 only.

Complete the following steps for each instance to refresh the instance control scripts and service names:

1. Uninstall the service.

```
instance/bin/httpdctl stop
```

```
instance/bin/httpdctl uninstall
```

2. Update the service with new features plus the revised `pivotal httpd instance` service names. The following command refreshes the `httpdctl` script for the instance. The command cannot be automated, because it is impossible to determine your instance locations.

```
./newserver --update --server=instance
```

3. Reinstall the service.

```
instance/bin/httpdctl install
```

```
instance/bin/httpdctl start
```

Update the Installation Path and Relocate All Instances

This option is available for all distributions, and is preferred as a longer term solution.

1. Complete the installation instructions to deploy Pivotal Web Server to your preferred location (typically `/opt/pivotal/webserver`). See *Installing Pivotal Web Server*.
2. Either copy or move the `/opt/vmware/vfabric-web-server/instance` tree to `/opt/pivotal/webserver/instance`, including all file permissions and ownership.

- Copy the tree to minimize downtime of active instances.
- If you move the tree instead, stop and uninstall `vfabric-web-server/instance` prior to moving the directory tree, as documented in 9 below.

3. Change to the `/opt/pivotal/webserver/instance` directory.

4. Invoke the `fixrootpath` as follows:

```
../fixrootpath.pl --srcdir=/opt/vmware/vfabric-web-server --dstdir=/opt/pivotal/webserver
```

5. Change to the `/opt/pivotal/webserver` directory.

6. Update the `httpdctl` script with new features plus the revised `pivotal httpd instance` service names:

```
./newserver --server=instance --update
```

7. **Optional.** Update the user and group names from `vfhttpd` to `pwshttpd` by modifying the `User` and `Group` directives in the `instance/conf/httpd.conf` file. Invoke the following commands from `/opt/pivotal/webserver/instance` to locate and replace the corresponding file and directory ownership:

```
find . -group vfhttpd | xargs chgrp pwshttpd
```

```
find . -user vfhttpd | xargs chown pwshttpd
```

8. Update the module and path names and the authorization sections in the `instance/conf/httpd.conf` file to match the 2.2 2.4 version. For guidance, see the *Apache HTTP Server documentation*.

9. Stop and uninstall the old instance if not performed before *step 2*. above:

```
/opt/vmware/vfabric-web-server/instance/bin/httpdctl stop
```

```
/opt/vmware/vfabric-web-server/instance/bin/httpdctl uninstall
```

10. Install and start the service from the new path:

```
instance/bin/httpdctl install
```

```
instance/bin/httpdctl start
```

11. Repeat steps *Step 2* through *Step 10* for each previously deployed server instance.

Migrating Enterprise Ready Server to Pivotal Web Server

This section has information about migrating Enterprise Ready Server (ERS) to Pivotal Web Server. For information about migrating ERS to Pivotal tc Server, see [Migrate an ERS Tomcat Instance to tc Server](#).

Pivotal currently provides two supported distributions of the open-source Apache HTTPD Server: Pivotal Enterprise Ready Server (ERS) and Pivotal Web Server. Pivotal Web Server is fully supported by Pivotal on common platforms such as Red Hat Enterprise Linux and Microsoft Windows, and support is available for additional, less-common platforms. See [Supported Configurations and System Requirements](#). Contact Pivotal support if your platform is not listed on the Pivotal Web Server product download page.

Pivotal Enterprise Ready Server is deprecated. Pivotal will no longer provide support for HTTPD version 2.0 as of November 30, 2013 and HTTPD version 2.2 as of November 30, 2014. For more information, see [VMware Support Policies](#). Please contact your account manager to learn about converting your existing active ERS licenses to Pivotal Web Server licenses.

This document describes how to migrate Enterprise Ready Server instances to use Pivotal Web Server. Although each product is based on the standard, open-source Apache HTTP Server, there are differences in Pivotal's implementations. See [Differences Between Pivotal Web Server and Pivotal ERS](#).

Note:

This document describes the migration procedure for Enterprise Ready Server based on Apache version 2.2.2.4. You can use the same procedures to migrate from Apache version 2.0, but be aware of additional changes. See [Upgrading to 2.4 from 2.2](#). See [Upgrading to 2.2 from 2.0](#).

Subtopics

[Preparing to Migrate](#)

[Migration Procedure](#)

Preparing to Migrate

Before you migrate, read the information in this section. You may also want to review [Differences Between Pivotal Web Server and Pivotal ERS](#).

Audit your Configurations

Migration from ERS to Pivotal Web Server is an opportune time for you to examine your Pivotal Web Server configurations for best practices and to update them where necessary. Note the following recommendations:

- Pivotal recommends that your deployment use the latest security modules and procedures.
- Pivotal recommends that you use the latest version of plug-in modules.

Pivotal does not guarantee compatibility of third-party plug-in modules between ERS and Pivotal Web Server. Pivotal recommends that you recompile third-party modules using the Developer build of Pivotal Web Server. You may need to contact your vendor to obtain the latest versions of the modules.

- Pivotal recommends that you use up-to-date SSL libraries and that you review your SSL cipher configurations and protocol directives.
- Pivotal recommends that you use modular configuration files.

Pivotal recommends that you use the `Include` directive to add external configurations to your `httpd.conf` file. When you use the `Include` directive to reference external configuration files, the `httpd.conf` file is more readable and your configurations are easier to manage. For example, the following directive references an external configuration file:

```
Include conf/extra/httpd-languages.conf
```

The default `httpd.conf` file that is created when you create a new Pivotal Web Server instance contains examples of including external configuration files. The external files are located in the `<instance>/conf/extra` directory) and are referenced with `Include` directives in the main `httpd.conf` file.

For more information, see the *Apache HTTPD* documentation.

Directory Structure Changes

The directory structure of a Pivotal Web Server installation closely matches a standard Apache 2.2.2.4 installation, whereas Pivotal ERS uses a more proprietary file-naming and directory convention. Migrating to Pivotal Web Server requires that you change configuration files and other scripts in your environment to reflect the new naming conventions. These changes make it easier to integrate third-party extensions with Pivotal Web Server.

Table 2: Directory Structure and File Name Changes describes the changes in directory structure, configuration file names, and script file names.

Table 2: Directory Structure and File Name Changes

Purpose	Enterprise Ready Server	Pivotal Web Server 5.5 6.2
Apache modules	<pre><install_dir>/httpd-2.2.2.4.<version>/modules/standard <install_dir>/httpd-2.2.2.4.<version>/modules/covalent <install_dir>/httpd-2.2.2.4.<version>/modules/jakarta <install_dir>/httpd-2.2.2.4.<version>/modules/perl <install_dir>/httpd-2.2.2.4.<version>/modules/php</pre>	<pre><install_dir>/httpd-2.2.2.4.<version>/modules</pre> <p>All modules reside in this single directory. Note that the <code>mod_perl</code> and <code>mod_php</code> modules are not included with Pivotal Web Server.</p> <p>See <i>Apache Module Changes</i>.</p>
Configuration file	<code><web_server_instance>/conf/httpsd.conf</code>	<pre><web_server_instance>/conf/httpd.conf</pre> <p>Additional configuration files for various Apache modules are located in the <code><web_server_instance>/conf/extras</code> subdirectory.</p>
Script for creating new Apache HTTPD server instances	Linux Perl script: <code><install_directory>/ers-server.pl</code>	Linux symlink to Perl script: <code><install_directory>/newserver</code>
	Windows Perl script: <code><install_directory>\ers-server.pl</code>	Windows PowerShell script: <code><install_directory>\newserver.ps1</code>

Purpose	Enterprise Ready Server	Pivotal Web Server 5.5 6.2
Start-up script	Linux: <code><web_server_instance>/bin/apache-startup.sh</code>	Linux: <code><web_server_instance>/bin/httpdctl</code>
	Windows: <code><web_server_instance>\bin\apache-startup.bat</code>	Windows PowerShell script: <code><web_server_instance>\bin\httpdctl.ps1</code>
Tools	<code><install_directory>/tools/*</code>	<code><install_directory>/httpd-<version>/bin/*</code>
Apache HTTPD binaries	HTTPD binaries are written to the installation directory. When the server is upgraded, the binaries are overwritten.	HTTPD binaries are written to a directory in the following format: <code>pivotal-web-server/httpd.- 2.2 2.4 .xx.x-bb.</code> (Where <code>xx.x</code> is the minor version number and <code>bb</code> is either 32 or 64 (for 32-bit or 64-bit versions)). When the server is upgraded, the binaries are not overwritten.
Windows service names	ERS <code><instance name></code> <code>httpsd</code>	Pivotal <code>httpd <instance name></code>

Apache Module Changes

The compiled Apache modules for Pivotal Web Server differ from those that are compiled and provided with Enterprise Ready Server. Pivotal recommends you substitute the modules described in [Table 3: Apache Modules](#). If necessary, you can compile and implement the modules yourself using the Developer build of Pivotal Web Server. To ensure compatibility, Pivotal recommends that you recompile all third-party modules. See [Complete Packages and Modules in Pivotal Web Server 5.5 6.2](#).

Table 3: Apache Modules

Module	Changes in Pivotal Web Server
<code>mod_perl</code>	Pivotal recommends that you use the <code>mod_fcgid</code> module to serve Perl-based applications. For more information, see the following article: FastCGI rocks mod_perl and FCGID .
<code>mod_php</code> (ERS)	The <code>mod_php</code> module is not included with Pivotal Web Server. Pivotal recommends that you use the <code>mod_fcgid</code> module to serve PHP-based applications. For more information on the benefits of using <code>mod_fcgid</code> , see Apache with fcgid: acceptable performance and better resource utilization .

Module	Changes in Pivotal Web Server
mod_snmp (ERS)	<p>Pivotal recommends that you use the Apache BMX framework to provide internal runtime information about your Pivotal Web Server instances to monitoring systems such as <i>Pivotal Hyperic</i>. The following modules are included and are enabled by default:</p> <ul style="list-style-type: none"> • mod_bmx • mod_bmx_status • mod_bmx_vhost <p>You can disable this functionality by commenting out the <code>LoadModule</code> directives for the BMX modules in the <code>httpd.conf</code> file.</p> <p>For more information, see <i>Configure BMX for Monitoring Pivotal Web Server Instances</i>.</p>
mod_bmx (Pivotal Web Server)	<p><code>mod_bmx</code> is a module in Pivotal Web Server. For a sample configuration, see the following directory in a newly created Pivotal Web Server instance:</p> <pre><instance>/conf/extras/conf/extra/httpd-info.conf</pre>
mod_jk	<p>Although <code>mod_jk</code> is still included with Apache version 2.2.24, it has limitations on header fields sizes, cannot be secured with SSL cryptography, and requires separate configuration of proxy worker pools.</p> <p>Pivotal recommends that you use <code>mod_proxy_balancer</code>, which supports both the <code>ajp</code> protocol used by <code>mod_jk</code>, and also the <code>http</code> and <code>https</code> protocol to legibly decipher the back-end connection stream and secure the back-end connections to Tomcat and tc Server. <code>mod_proxy_balancer</code> is configured using the <code>httpd.conf</code> file syntax.</p> <p>For <code>mod_proxy</code> configuration options, see http://httpd.apache.org/docs/2.2/2.4/mod/mod_proxy.html.</p> <p>For configuration instructions, see <i>Configure Load Balancing Between Two or More tc Runtime Instances</i>.</p>
mod_version	<p>The <code>mod_version</code> module is compiled into Pivotal Web Server. If your ERS <code>httpdsd.conf</code> file contains a <code>LoadModule</code> line, delete the line. The functionality of <code>mod_version</code> is still available in your migrated Pivotal Web Server instances.</p> <p>For example, delete the following line:</p> <pre>LoadModule version_module <path></pre>

Module	Changes in Pivotal Web Server
mod_ssl	<p>In ERS instances, you conditionally configure SSL in the <code>httpdsd.conf</code> file by wrapping the <code>LoadModule mod_ssl</code> directive within an <code><IfDefine SSL></code> directive. When you start the server with the <code>startssl</code> command, SSL is enabled.</p> <p>In Pivotal Web Server, you configure SSL in the <code>httpd.conf</code> file by adding the <code>LoadModule ssl_module <path></code> line without a conditional <code><IfDefine SSL></code> directive. When you start the server instance using the <code>start</code> command, the server starts using SSL.</p> <p>See <i>Startup Script Changes</i>.</p> <p>Pivotal Web Server now uses Open SSL 1.01 and supports Transport Layer Security (TLS) 1.1 and 1.2.</p>

Startup Script Changes

The name of the script that you use to start Pivotal Web Server instances has changed, and has some new options. In addition, configuration of the multi-process module (MPM) has changed. The startup scripts are located in the `bin` subdirectory of a server instance.

Table 4: Startup Script Changes

Item	Enterprise Ready Server	Pivotal Web Server 5.5 6.2
Startup (control) script name	Linux: <code><web_server_instance>/bin/apache_startup.sh</code>	Linux: <code><web_server_instance>/bin/httpdctl</code>
	Windows: <code><web_server_instance>\bin\apache_startup.bat</code>	Windows: <code><web_server_instance>\bin\httpdctl.ps1</code>
Startup (control) script parameters	<p>The following commands are available with the <code>apache_startup</code> script:</p> <ul style="list-style-type: none"> • <code>start</code> • <code>startssl</code> • <code>stop</code> • <code>restart</code> • <code>graceful</code> • <code>configtest</code> • <code>configtestssl</code> 	<p>The following commands are available with the <code>httpdctl</code> script:</p> <ul style="list-style-type: none"> • <code>start</code> • <code>stop</code> • <code>restart</code> • <code>gracefulstop</code> • <code>graceful</code> • <code>status</code> • <code>install</code> • <code>uninstall</code> • <code>configtest</code> <p>See <i>httpdctl Reference</i>.</p>

Item	Enterprise Ready Server	Pivotal Web Server 5.5 6.2
Startup script command with SSL enabled	To use SSL in an ERS instance you configure and load an SSL module in the <code>httpsd.conf</code> file, and you use the <code>startssl</code> option when you run the <code>apache_startup</code> script.	To use SSL in a Pivotal Web Server instance, you configure a SSL module in the <code>httpd.conf</code> file and use the regular <code>start</code> option when you run the <code>httpdctl</code> script.
Multi-process module (MPM) configuration	You configure MPM in the <code>startup.properties</code> file.	<p>You configure MPM when you create a Pivotal Web Server instance. Use the following option of the <code>newserver</code> command:</p> <pre>--mpm=worker prefork event</pre> <p>(The default value is <code>worker</code>.)</p> <p>See <i>newserver Prompts and Command Reference</i>.</p> <p>If you need to modify MPM after creating a server, change the value of the <code>default_mpm</code> variable to <code>worker</code>, <code>prefork</code>, or <code>event</code> by changing the following line in the <code>httpdctl</code> script:</p> <pre>default_mpm="worker"</pre>

Migration Procedure

To migrate an Enterprise Ready Server installation to a Pivotal Web Server installation:

1. Install Pivotal Web Server on each host where you will run a Pivotal Web Server instance. See *Installing Pivotal Web Server*.
2. Create a new Pivotal Web server instance. See *Creating and Using Pivotal Web Server Instances*.

Note:

If you specified `mpm` options in the Enterprise Ready Server `startup.properties` file, you must specify the `--mpm=worker|prefork|event` option when creating the new Web server instance. See *Startup Scripts* and *newserver Prompts and Command Reference*.

3. Make a backup copy of the default configuration file from the new Pivotal Web Server instance. The file is in the following location:

```
<Web Server instance>/conf/httpd.conf.
```

4. Copy the following file from the Enterprise Ready Server instance directory:

```
servers/<Web server instance>/conf/httpsd.conf
```

into the `conf` directory of your new Pivotal Web Server instance and rename the file to `httpd.conf`.

5. Open the newly-copied `httpd.conf` configuration file in a text editor and modify all of the paths to point to the new Web server instance. Make sure to change the following paths:

- `ServerRoot` property.
- Apache modules—modify the appropriate `LoadModule` directives.
- `DocumentRoot` property.
- `<Directory>` statements.

- Include directives.
 - Paths to any scripts referenced in the configuration.
 - Paths to any log files referenced in the configuration.
6. If your Pivotal Web server instance uses any of the Apache modules that have changed, change the `LoadModule` directives to point to the new modules. Review the changes listed in *Apache Modules Changes* for any changes that apply to your deployment.
 7. Move static content from the previous Document Root directory to the new Document Root (or change `DocumentRoot` to point to the correct location.)
 8. Make sure that the user and group name used to run Pivotal Web Server instances exist on the host system. Use the host operating system commands to create the required user and group as indicated in the following table.

The default user and group names used to run the `httpd` server have changed.

Table 5: User and group name changes

Default Name	Enterprise Ready Server	Pivotal Web Server
Default group name	nobody	pwshttpd
Default user name	nobody	pwshttpd

9. Review the configuration file for any modules that you are not using in your deployment, and remove any `LoadModule` directives for these modules.
10. If you are using SSL, move the certificates and key file to your new Pivotal Web Server instance directory. Make sure the permissions allow access to these files. (The `root` user should have read-only access.)
11. If a previous instance of Enterprise Ready Server was running on the host computer, make sure you uninstall the obsolete service and install the new Pivotal Web Server instance as a service. See *Windows: Start and Stop Pivotal Web Server Instances* or *Installing Pivotal Web Server Instances as Unix Services*.
12. Start the Pivotal Web Server instance, using the `httpdctl` script. See *Startup Script Changes*.
13. Test the server using a Web browser.

Creating and Using Pivotal Web Server Instances

To start using Pivotal Web Server, you explicitly create a new instance after you install it. An instance is not created for you by default.

Subtopics

Description of Pivotal Web Server Instances

Create Pivotal Web Server Instances

newserver Prompts and Command Reference

Unix: Start and Stop Pivotal Web Server Instances

Windows: Start and Stop Pivotal Web Server Instances

httpdctl Reference

Serve a Sample HTML File from Your Pivotal Web Server Instance

Description of Pivotal Web Server Instances

A Pivotal Web Server *instance* is a complete, discrete HTTP server configuration.

You can create multiple instances that you can run simultaneously on the same computer if you do not use the same ports in two different instances. For example, the default HTTP listening port on Unix is 80, and only one instance on any computer is allowed to communicate on port 80 at any one time. So if you wanted to have two Pivotal Web Server instances running at the same time on the same Unix computer, you configure one instance to use a port other than 80.

After you create an instance, its corresponding directory contains subdirectories that in turn contain all the data required to run a given Pivotal Web Server instance. This data includes configuration information and all other data that is associated with that instance's configuration. For example, assume you installed Pivotal Web Server in `/opt/pivotal/webserver` and create an instance called `myserver`:

```
prompt$ cd /opt/pivotal/webserver/myserver
prompt$ ls
bin  cgi-bin  conf  ftpdocs  htdocs  logs  proxy  ssl  var
```

The `conf` directory contains the Pivotal Web Server configuration files, such as `httpd.conf`. The `bin` directory contains the startup script used to start and stop the `myserver` instance (`httpdctl`). Each of these directories is specific to the `myserver` instance. Each instance you create will have a similar set of directories.

Create Pivotal Web Server Instances

Create Instances Using the `newserver` Command

You create a new Pivotal Web Server instance with the `newserver` command. The command creates a new directory that contains the instance-specific configuration files.

The `newserver` command format depends on your operating system:

- `newserver` : Perl script for Unix operating systems.
- `newserver.ps1`: Powershell script for Windows operating systems.

The command-line options for the two flavors are exactly the same. Where appropriate, the following procedure calls out the different usage depending on whether you are on Unix or Windows.

Prerequisites

- Complete the appropriate procedure in *Installing Pivotal Web Server*.
- As of version 5.2 of Pivotal Web Server, you must use Windows PowerShell to execute Pivotal Web Server scripts on Windows computers. See the prerequisites section of *Windows: Install Pivotal Web Server from a ZIP File* for information on installing PowerShell (if it is not already installed on your Windows computer) and enabling it for script processing.

Procedure

1. Log on to your computer as `root` (Unix) or the Administrator user (Windows) and start a terminal (Unix) or PowerShell window (Windows).

To start a PowerShell window, go to Start > All Programs > Accessories > Windows PowerShell, then right-click on Windows PowerShell and select Run as Administrator.

2. Change to the directory in which you installed Pivotal Web Server. For example, on Unix:

```
prompt# cd /opt/pivotal/webserver
```

3. Run the `newserver` command to create the new instance; the command prompts you for information about the new server.

The only required command option is `--server`, with which you specify the name of your Pivotal Web Server instance. On Unix, use the Perl flavor; for example:

```
prompt# ./newserver --server=myserver
```

On Windows, use the PowerShell script:

```
PS prompt> .\newserver.ps1 --server=myserver
```

In both preceding examples, the way you specify the options is exactly the same. In the examples, the new instance is called `myserver` and its server directory is `/opt/pivotal/webserver/myserver`.

For additional options, see *newserver Prompts and Command Reference*.

4. Enter values for the `newserver` prompts as the command requests information about your new instance. You can use the default values for many of the prompts, or even leave them blank.

newserver Prompts and Command Reference provides additional information about the prompts.

What to do next

- Start the Pivotal Web Server instance and verify that it is working correctly. See *Unix: Start and Stop Pivotal Web Server Instances* and *Windows: Start and Stop Pivotal Web Server Instances*.
- Complete the procedure in *Serve a Sample HTML File from your Pivotal Web Server Instance*.
- Configure your instance as described in *Configuring Pivotal Web Server*.

Windows: Create and Modify Instances

The MSI installer for Microsoft Windows installs the Pivotal Web Server Configurator with the application.

Creating a Web Server Instance

1. Click the **Start** button > **All Programs** > **Pivotal** > **Pivotal Web Server Configurator**.

The application opens and displays the list of existing Web Server instances.

2. Click the **Add** button.

The **Basic Settings** window opens.

3. Complete the following fields:

- Enter a server name. The server name identifies the Web Server instance locally.
- Enter the port number for HTTP listening.

- Enter the host name. The host name identifies the web server instance on your network or the Internet.
 - Accept the default server location, or browse to and select a new location.
4. Click the **Next** button.

The **SSL Settings and Administrator Email** page opens.

5. **Optional.** Enable SSL:

- Select the **Enable SSL** checkbox if you are enabling SSL.
- Enter the port for HTTPS listening. The default value is 443.
- Browse to and select a SSL certificate file location. For information about configuring SSL between Pivotal Web Server and Pivotal tc Server, see *Configure SSL Between Pivotal Web Server and Pivotal tc Server*.
- Browse to and select a key file location.

If you do not want to enable SSL, then leave the checkbox cleared.

6. Enter the Web Server administrator email address.
7. Click the **Next** button.
8. Review the installation configuration list.
9. Click the **Finish** button when you are ready to proceed.

Note: The configurator does not display a progress bar as it creates the new instance.

The configurator displays the new Web Server instance in the list.

10. Click the **Install** button to create a Windows service for the instance.

Starting and Stopping a Web Server Instance

1. Click the **Start** button > **All Programs** > **Pivotal** > **Pivotal Web Server Configurator**.

The application opens and displays the list of existing Web Server instances.

2. Select the instance that you want to start or stop.
3. Start or stop the instance.
 - Click the **Start** button to start the Windows service for the instance.
 - Click the **Stop** button to stop the Windows service for the instance.

Removing a Web Server Instance

1. Click the **Start** button > **All Programs** > **Pivotal** > **Pivotal Web Server Configurator**.

The application opens and displays the list of existing Web Server instances.

2. Select the instance that you want to remove.
3. Click the **Remove** button.

The configurator stops the instance Windows service, if started, and removes the instance from your hard drive.

newserver Prompts and Command Reference

The `newserver` command has a number of options and prompts, as described in the two tables that follow.

The `newserver` command format depends on your operating system:

- `newserver` : Perl script for Unix.
- `newserver.ps1`: PowerShell script for Windows.

The command-line options for the two flavors are exactly the same.

Table 6: Options of the newserver Command

Option	Description	Required?
<code>--server=servername</code>	<p>Name of the new Pivotal Web Server instance. The value of this option becomes the name of the directory that contains the instance configuration files, and by default is the name of the host.</p> <p>The value of <code>servername</code> must be a valid DNS value and consist only of ASCII letters, digits or the dash character. Illegal characters include but are not limited to control characters below ASCII 32 as well as the following symbols: <code>< > : " / \ ? *</code>.</p> <p>If you want to use an internationalized (i18n) name for the instance directory, you must enter the correct Punycode domain name provided by the registrar for the actual hostname. Do this by either specifying the <code>--set HostName=punycode-hostname</code> option at the command-line or entering the value interactively when the <code>newserver</code> command prompts you for the hostname.</p>	Yes.
<code>--httpddir=httpddir</code>	<p>Directory that contains the Apache HTTP binaries.</p> <p>The default value is <code>rootdir/httpd-httpdver</code>, such as <code>/opt/pivotal/webserver/httpd-2.2.29.0-2.4.10-64</code>.</p>	No.
<code>--httpdver=httpdver</code>	<p>Version of the Apache HTTP binaries you want your instance to use.</p> <p>The default value is <code>2.2.2.4</code>, which is a symbolic link to the actual installed version of the binaries, such as <code>2.2.29.0-2.4.10</code>.</p>	No.

Option	Description	Required?
<code>--mpm=<i>mpm</i></code>	<p>Specifies the type of multi-processing module (MPM) that the instance uses. Valid values are:</p> <ul style="list-style-type: none"> <code>worker</code>: Threaded MPM, ideal if you need a great deal of scalability. By using threads to serve requests, the instance can serve many requests with fewer system resources than a process-based server. <code>prefork</code>: Non-threaded, pre-forking MPM if you require stability or compatibility with older software. <code>event</code>: Less proven but higher-efficiency asynchronous connection-keepalive MPM. The <code>event</code> MPM offers little benefit for HTTPS connections, but is able to handle more simultaneous kept-alive and pending HTTP connections. <p>The default value is <code>worker</code>.</p>	No.
<code>--overlay</code>	Specifies that, if <code>serverdir</code> exists, you want to overwrite the existing files with new ones.	<p>No.</p> <p>If you do not specify this option, and <code>serverdir</code> exists, the <code>newserver</code> command returns an error and suggests you specify a unique name and directory location for the new instance.</p>
<code>--quiet</code>	Specifies that the <code>newserver</code> command should use default values for all prompts.	<p>No.</p> <p>If you do not specify this option, <code>newserver</code> interactively prompts for all answers.</p>
<code>--rootdir=<i>rootdir</i></code>	<p>Directory that contains the <code>httpd-2.2.2.4.version</code> directory, which in turn contains the Apache HTTP binaries.</p> <p>The default value is the current directory.</p>	No.
<code>--serverdir=<i>serverdir</i></code>	<p>Directory in which you want the new instance directory to be created.</p> <p>The default value is <code>rootdir</code>.</p>	No.

Option	Description	Required?
<code>--set token=value</code>	<p>Specifies one or more tokens for which you would like to specify a custom value. The tokens are variables in the templates used to create new Pivotal Web Server instances and correspond to a directive, or part of a directive, in the new instance's configuration. Each available token has a default value (listed below) that is automatically configured if you do not override it using the <code>--set</code> option.</p> <p>You can specify the following tokens; note that the token names are case-sensitive:</p> <ul style="list-style-type: none"> • User: User that the Pivotal Web Server processes run as. Corresponds to the <code>User</code> directive in <code>conf/httpd.conf</code>. Default value is <code>pwshttpd</code>. • Group: Group to which the user who runs the Pivotal Web Server processes belongs. Corresponds to the <code>Group</code> directive in <code>conf/httpd.conf</code>. Default value is <code>pwshttpd</code>. • Port: HTTP port that the Pivotal Web Server instance listens to. Corresponds to the port number in the <code>Listen</code> directive in <code>conf/httpd.conf</code>. Default value is 80. • SSLPort: HTTPS port that the Pivotal Web Server instance listens to for secure communications. Corresponds to the port number in the <code>Listen https</code> directive in <code>conf/extras/httpd-ssl.conf</code>. Default value is 443. • HostName: Name of the host that the instance uses to identify itself. Corresponds to the hostname part of the <code>ServerName</code> directive in the <code>conf/httpd.conf</code> file. Default value is the value you specified for the required <code>--server</code> option. 	No.

Option	Description	Required?
	<ul style="list-style-type: none"> ServerAdmin: Email address of the administration user who should get emails when there are problems with the instance. Corresponds to the <code>ServerAdmin</code> directive in the <code>conf/httpd.conf</code> file. Default value is <code>webmaster@HostName..</code> <p>The following example shows how to specify that the new Pivotal Web Server instance run as the <code>newhttpd</code> user in the <code>newhttpd</code> group:</p> <pre>prompt# ./newserver -- server=myserver --set User=newhttpd --set Group=newhttpd</pre>	
<code>--sourcedir=sourcedir</code>	<p>Name of the directory that contains the template that <code>newserver</code> uses to create the new Pivotal Web Server instance.</p> <p>The default value is <code>httpdir/_instance</code>.</p>	No.

Table 7: newserver Prompts

Prompt	Description
<i>Enable SSL and create a default key [y/n]?</i>	Enabling SSL provides secure communication between client and server by allowing mutual authentication; the use of digital signatures for integrity; and encryption for privacy. If you answer yes, you are later asked for information that will be used to create a certificate.
<i>Server hostname (e.g. www.example.com) [myserver]?</i>	Name that the Pivotal Web Server instance uses to identify itself. If your host does not have a registered DNS name, enter its IP address. The default value is the value you entered for the <code>--server</code> option.
<i>Administrator email [webmaster@myserver]?</i>	Email address to which Pivotal Web Server instances send problems. This address appears on some instance-generated pages, such as error documents.
<i>Port for http:// traffic [80]?</i>	HTTP port to which the Pivotal Web Server instance listens. Default value is 80 when running the <code>newserver</code> command as the <code>root</code> user on Unix, 8080 otherwise.

Prompt	Description
<i>Port for https:// SSL traffic [443]?</i>	HTTPS port to which the Pivotal Web Server instance listens. Default value is 443 when running the <code>newserver</code> command as the <code>root</code> user on Unix, 8443 otherwise.
If you previously specified that you want to enable SSL...	<p>The <code>newserver</code> command prompts you for information required to create the private key, such as the size of the SSL RSA key in bits and the PEM pass phrase you specify when you start the instance.</p> <p>You also are prompted to enter information for your certificate. The information is mostly about your Distinguished Name, or DN, that will be incorporated into your certificate request. As indicated, some fields have default values. You can also leave some fields blank by entering a '.' (period.)</p> <p>When <code>newserver</code> completes, it generates the following SSL files in the <code>ssl</code> subdirectory of the instance directory:</p> <ul style="list-style-type: none"> <code>instance_name.key</code>: Unencrypted private key. The file has a permission code of 0600 for additional security. <code>instance_name.pem</code>: DES 3 encrypted private key. <code>instance_name.csr</code>: Certificate-signing request. Submit this file to the Certificate Authority. <code>instance_name.crt</code>: Self-signed certificate. Replace this certificate with a signed certificate by the CA. <p>Important: Be sure to record the passphrase to decrypt the *.pem file and back up the file. Never transmit the .key file or cause it to be readable by others.</p>

Unix: Start and Stop Pivotal Web Server Instances

You interactively start, stop, or restart a Pivotal Web Server instance on Unix with the `httpdctl` shell script in the `bin` directory of the instance.

Warning: You always use the start script in the `bin` of the instance directory, such as `/opt/vmware/pivotal-web-server/myserver/bin`. Do not use the start script in the `httpd- 2.2 2.4 /bin` sub-directory of the main installation directory.

You can also install a Pivotal Web Server instance as a Unix service so it automatically starts and stops when the operating system itself is started and stopped. If you install it as a service, you can also start and stop the service by using the `/etc/init.d/service-name` script. For more information, see *Installing Pivotal Web Server Instances as Unix Services*.

Prerequisites

- Complete the appropriate procedure in *Create Pivotal Web Server Instances*.

Procedure

1. Log in to your Unix computer as the `root` user.
2. Start a terminal window and change to the `bin` sub-directory of your Pivotal Web Server instance's root directory. For example, if you created an instance called `myserver` that lives in the installation directory `/opt/pivotal/webserver`:

```
prompt# cd /opt/pivotal/webserver/myserver/bin
```

3. Start the instance using the `./httpdctl start` command:

```
prompt# ./httpdctl start
```

You should see a message as follows:

```
Starting Apache:
Server started OK
```

4. To test that the Pivotal Web Server instance actually started, navigate to the `http://host:port` URL in your browser, where `host` refers to the host computer (you can use `localhost` if your browser is on the same computer) and `port` refers to the HTTP listen port number you provided when you created the instance. The default value is `80`

For example, if you are using the default ports on your local computer, you can use this URL:

```
http://localhost:80
```

If the instance started successfully, you should see the Welcome page.

5. To get status about the instance:

```
prompt# ./httpdctl status
```

6. To stop the instance immediately, even if there are current connections in use:

```
prompt# cd /opt/pivotal/webserver/myserver/bin
prompt$ ./httpdctl stop
```

To stop the instance gracefully:

```
prompt$ ./httpdctl gracefulstop
```

See [httpdctl Reference](#) for the full list of available `httpdctl` commands.

What to do next

- Complete the procedure in [Serve a Sample HTML File from your Pivotal Web Server Instance](#).
- Configure your instance as described in [Configuring Pivotal Web Server](#).

Installing Pivotal Web Server Instances as Unix Services

You can install a Pivotal Web Server instance as a system service on Unix, Linux, so it automatically starts and stops when the operating system itself starts and stops. If you install it as a service, you start and stop the service by using the `/etc/init.d/ServiceName` script.

Procedure

1. Log in to your computer as the `root` user.
2. Start a terminal window and change to the `bin` sub-directory of your Pivotal Web Server instance's root directory. For example, if you created an instance called `myserver` that lives in the installation directory `/opt/pivotal/webserver`:

```
prompt# cd /opt/pivotal/webserver/myserver/bin
```

3. Install the instance as a service using the `./httpdctl install` command:

```
prompt# ./httpdctl install
```

You should see the following output:

```
Installing Pivotal httpd myserver
```

```
as the unix service pivotal-httpd-myserver
```

The display name for the service name is `pivotal httpd instance_name` . The service name is the display name, with spaces replaced by dashes: `pivotal-httpd-instance_name` .

4. After creating the service, you can control it (1) with the system-config-services GUI utility, (2) by running the `bin/httpdctl` shell script from the instance directory, or (3) by running the `/etc/init.d/ServiceName` script, passing it the same commands as `httpdctl`. For example, to start the `myserver` instance:

```
prompt# /etc/init.d/pivotal-httpd-myserver start
```

5. To uninstall the instance as a Unix service:

```
prompt# ./httpdctl uninstall
```

Windows: Start and Stop Pivotal Web Server Instances

You start, stop, or restart a Pivotal Web Server instance on Windows by first installing it as Windows service using the `httpdctl.ps1` PowerShell script in the `bin` directory of the instance directory, and subsequently using the Windows Services console to start or stop it.

Warning: You always use the start script in the `bin` of the instance directory, such as `c:\opt\vmware\pivotal-web-server\myserver\bin`. Do not use the start script in the `httpd- 2.2 2.4 \bin` sub-directory of the main installation directory.

Prerequisites

- Complete the appropriate procedure in *Create Pivotal Web Server Instances*.
- As of version 5.2 of Pivotal Web Server, you are required to use Windows PowerShell to execute Pivotal Web Server scripts on Windows computers. See the prerequisites section of *Windows: Install Pivotal Web Server from a ZIP File* for information on installing PowerShell (if it is not already installed on your Windows computer) and enabling it for script processing.

Procedure

1. Log in to your Windows computer as the Administrator user and start a PowerShell window by going to Start > All Programs > Accessories > Windows PowerShell, then right-clicking on Windows PowerShell and selecting Run as Administrator.
2. Change to the `bin` subdirectory of the root directory for the Pivotal Web Server instance.

For example, if you created an instance called `myserver` that lives in the installation directory `c:\opt\vmware\pivotal-web-server`:

```
PS prompt> cd c:\opt\vmware\pivotal-web-server\myserver\bin
```

3. Install the instance as a Windows service by running the `httpdctl.ps1 install` command:

```
PS prompt> .\httpdctl.ps1 install
```

The display name for the service name is `pivotal httpd instance_name` . The service name is the display name, with spaces removed: `pivotal-httpd instance_name` .

After installing the service, you can control and further configure the service in several ways: (1) `httpdctl` command options, (2) the Windows Services console, or (3) the `sc` command

4. To test that the Pivotal Web Server instance actually started, navigate to the `http://host:port` URL in your browser, where `host` is the host computer (you can use `localhost` if your browser is on the same computer), and `port` is the HTTP port number you provided when you created the instance. The default value on Windows is 80.

For example, if you are using the default ports on your local computer, use this URL:

```
http://localhost:80
```

If the Pivotal Web Server instance started successfully, you should see the Welcome page.

5. To get status about a running Pivotal Web Server instance, execute the following command:

```
PS prompt> .\httpdctl.ps1 status
```

6. To uninstall the Pivotal Web Server instance as a Windows service, use the following command:

```
PS prompt> cd c:\opt\vmware\pivotal-web-server\myserver\bin
PS prompt> .\httpdctl.ps1 uninstall
```

See *httpdctl Reference* for the full list of available `httpdctl` commands.

What to do next

- Complete the procedure in *Serve a Sample HTML File from your Pivotal Web Server Instance*.
- Configure your instance as described in *Configuring Pivotal Web Server*.

httpdctl Reference

Use the `httpdctl` script to control a Pivotal Web Server instance: start 'n' stop it, install it as a service, and so on. The script is located in the `bin` directory of the instance directory, such as `/opt/pivotal/webserver/myserver/bin`.

The `httpdctl` script format depends on your operating system:

- `httpdctl`: Perl script for Unix. See *Unix: Start and Stop Pivotal Web Server Instances* for usage examples.
- `httpdctl.ps1`: PowerShell script for Windows. See *Windows: Start and Stop Pivotal Web Server Instances* for usage examples.

Script commands are the same for both. Commands are described in the following table.

Table 8: httpdctl Script Commands

Command	Description
<code>start</code>	Starts the Pivotal Web Server instance. If the instance is already running, the command returns an error.
<code>stop</code>	Forcibly stops the Pivotal Web Server instance. All currently opened connections are aborted.
<code>gracefulstop</code>	Gracefully stops the Pivotal Web Server instance, which means that the script waits until all currently open connections are closed rather than aborting them forcibly.
<code>restart</code>	Restarts the instance. If the instance was not originally running, the script starts it. If the instance was not originally running, the script starts it. The script also runs a <code>configtest</code> before starting the instance.

Command	Description
<code>graceful</code>	Gracefully restarts the instance. If the instance is not running, it is started. This command differs from a normal restart in that currently open connections are not aborted. A side effect is that old log files will not be closed immediately. This means that if you use this command in a log rotation script, a substantial delay may be necessary to ensure that the old log files are closed before processing them. This command runs a <code>configtest</code> before starting the instance.
<code>status</code>	Displays basic status information about the instance, such as whether it is running and its process id (PID) if so.
<code>install</code>	<p>Installs the instance as a service on Windows, Unix, and Linux.</p> <p>After installing the service on Windows, you use the Windows Services console or the <code>sc</code> command to start, stop, and restart the Pivotal Web Server instance, and configure whether the service starts automatically when Windows starts, and so on. For more information about the <code>sc</code> command, see http://technet.microsoft.com/en-us/library/bb490995.aspx</p> <p>On Unix, the instance is installed as a script file in the <code>/etc/init.d</code> directory with name <code>pivotal-httpd-<i>instance_name</i></code>. The service automatically starts and stops when Unix is started or stopped.</p>
<code>uninstall</code>	Uninstalls the instance as Windows, Unix, or Linux service. On Windows, the instance is removed from the Service registry. On Unix, the command deletes the <code>/etc/init.d/pivotal-httpd-<i>instance_name</i></code> script file.
<code>configtest</code>	Runs a syntax test against the configuration files, such as <code>conf/httpd.conf</code> . The script parses the configuration files and either reports Syntax OK or outputs detailed information about the particular syntax error.

Serve a Sample HTML File from Your Pivotal Web Server Instance

After you install Pivotal Web Server and create an instance, you can use it to host your entire Web site. This section does not describe the entire process; rather, it simply shows how to serve an HTML file from the default document root of your instance.

Prerequisites

- Create and start a Pivotal Web Server instance. See *Create Pivotal Web Server Instances*.
- Create or download one or more sample HTML pages that you want to serve from the instance.

Procedure

1. Open the configuration file for your Pivotal Web Server instance and make note of the value of the `DocumentRoot` directive, which is the directory out of which the instance serves your documents. By default, Pivotal Web Server takes all requests from this directory.

The configuration file is called `httpd.conf` and is located in the `INSTANCE-DIR/conf`, such as `/opt/pivotal/webserver/myserver/conf/httpd.conf`. The `DocumentRoot` directive looks like the following:

```
DocumentRoot "/opt/pivotal/webserver/myserver/htdocs"
```

2. Copy your sample HTML pages to the document root.

For example, if you have a `hello.html` page in the `/home/samples` directory that you want to serve up:

```
prompt# cp /home/samples/hello.html /opt/pivotal/webserver/myserver/htdocs
```

3. Invoke the HTML page in your browser using the Pivotal Web Server instance.

For example, if your browser is running on the same computer as Pivotal Web Server and the instance is listening at the default port 80, the URL is as follows:

```
http://localhost/hello.html
```

Because the instance is using the default port of 80, you do not have to explicitly specify it in the URL. If you set a different port, such as 8000, then the URL would be:

```
http://localhost:8000/hello.html
```

You should see your `hello.html` page in your browser.

4. You can create a directory hierarchy under the document root to better organize your HTML pages.

For example:

```
prompt# cd /opt/pivotal/webserver/myserver/htdocs
prompt# mkdir fun
prompt# cp /home/samples/hello.html fun
```

The URL to invoke the HTML page would now be:

```
http://localhost/fun/hello.html
```

What to do next

- Configure Pivotal Web Server instances to take advantage features such as load balancing, virtual hosts, and SSL. See *Configuring Pivotal Web Server Instances*.

Configuring Pivotal Web Server Instances

The default configuration of a newly created Pivotal Web Server instance is fairly simple. Although the configuration is likely adequate for your needs, sometimes you might need to further configure the instance to enable one of its many useful features, such as load-balancing between two or more tc Runtime instances. This chapter provides some information to get you started.

For complete documentation on how to configure Pivotal Web Server instances, see [Apache HTTP Server Version 2.2 Documentation](#)[Apache HTTP Server Version 2.4 Documentation](#). Because Pivotal Web Server is based on Apache HTTP server, the general configuration documentation on the Apache Web site applies to Pivotal Web Server as well.

Subtopics

[Using the Sample Configuration Files to Enable Features and Modify Configuration](#)

[Configure Load Balancing Between Two or More tc Server Instances](#)

[Configure SSL Between Pivotal Web Server and Pivotal tc Server](#)

[Configure BMX for Monitoring Pivotal Web Server Instances](#)

[Metrics](#)

Using Sample Configuration Files to Enable Features and Modify Configuration

All Pivotal Web Server instances include sample configuration files that you can use to enable extra features in the server instance or to modify its default configuration. These files are located in the `INSTANCE-DIR/conf/extra` directory, where `INSTANCE-DIR` refers to the instance directory, such as `/opt/pivotal/webserver/myserver`.

For example, the `httpd-info.conf` sample configuration file shows how you can get information about the requests being processed by the Pivotal Web Server instance as well as information about the configuration of the instance. The `httpd-ssl.conf` file shows how to provide SSL support. It contains the configuration directives to instruct the instance how to serve pages over an HTTPS connection.

For your convenience, the main Pivotal Web Server configuration file for a particular instance (`INSTANCE-DIR/conf/httpd.conf`) already includes commented-out lines for including each sample configuration file. For example, the line to include the `httpd-info.conf` configuration file is as follows:

```
#Include conf/extra/httpd-info.conf
```

To include the configuration file, simply uncomment the `Include` directive:

```
Include conf/extra/httpd-info.conf
```

You do not have to use `Include` in this way; you can simply copy and paste the information in a sample configuration file into the main configuration file.

The sample configuration files are full of comments on how exactly to enable the feature they configure. Be sure to read these comments before you proceed further.

What to do next

- Restart the Pivotal Web Server instance for the configuration changes to take effect. For example, on Unix:

```
prompt# cd /opt/pivotal/webserver/myserver
prompt# bin/httpdctl restart
```

Configure Load Balancing Between Two or More tc Runtime Instances

You can configure a Pivotal Web Server instance to perform simple load balancing between two or more tc Runtime instances.

In the procedure that follows, you configure a Pivotal Web Server instance to run *in front* of the tc Runtime instances; this Pivotal Web Server instance receives all requests from users, and then passes them back to the tc Runtime instances using a specified load-balancing algorithm. Responses from the tc Runtime instances are then routed back through this same Pivotal Web Server instance. For this reason, the Pivotal Web Server instance acts like a proxy (both reverse and forward) so that the users never know the URLs of the backend tc Runtime instance that are actually doing all the work. Additionally, the Pivotal Web Server instance ensures that the load on each tc Runtime instance is balanced. You can specify that each tc Runtime instance take on an equal work load, or you can specify that one instance work twice as hard as the others.

In the procedure, the following scenario pertains. These assumptions are *not* requirements; your environment might be very different. The assumptions are listed only to make the procedure easier to understand.

- Two tc Runtime instances are running at the following two hosts and port numbers:
 - `http://192.168.0.203:8081`
 - `http://192.168.0.203:8082`

The two tc Runtime instances are running on the same computer, are part of the same installation and their respective `CATALINA_BASE` variables are as follows:

- `/var/opt/pivotal/pivotal-tc-server-standard/instanceOne`
- `/var/opt/pivotal/pivotal-tc-server-standard/instanceTwo`
- Each tc Runtime instance is configured exactly the same (other than the value of the various ports).
- You have deployed the same application to both tc Runtime instances and the URL context is the same in both instances: `/my-app`.
- You want all users of the application to first go through the front-end Pivotal Web Server instance, and any evidence of the backend tc Runtime instances upon which the application is actually deployed should be hidden from the user.
- Pivotal Web Server is installed on a different computer than Pivotal tc Server. The name of the particular Pivotal Web Server instance is `lb-server` and its home directory is `/opt/pivotal/webserver/lb-server`.
- You want to configure *sticky sessions*, which means that the Pivotal Web Server instance always routes the requests for a particular session to the same tc Runtime instance that serviced the first request for that session.
- You want to use the HTTP protocol for all communication between the Pivotal Web Server and the tc Runtime instances.

The load balancing described in this procedure is very simple, although you have many options available to further customize it. At appropriate locations in the procedure, links to the Apache HTTP Server documentation are provided for additional configuration options not covered by this specific scenario. Adapt the procedure for your particular environment.

As part of the procedure, you update the configuration files of both the Pivotal Web Server instance and the two tc Runtime instances.

Prerequisites

- *Install Pivotal Web Server* on your platform and *create a new instance*.

- Install Pivotal tc Server on the same or different computer as Pivotal Web Server, and create two more instances. Make note of the host and port numbers of the two instances. See the Pivotal tc Server documentation for details.
- Deploy the same application to the two tc Runtime instances.

Procedure

To configure load balancing for the scenario described in the introduction to this section, follow these steps:

1. On the computer on which Pivotal Web Server is installed, stop the instance, if it is currently running. Following the example and assumptions:

```
prompt# cd /opt/pivotal/webserver/lb-server
prompt# bin/httpdctl stop
```

2. Open the `httpd.conf` configuration file of the Pivotal Web Server instance and ensure that the three required `LoadModule` directives (`proxy_balancer_module`, `mod_proxy`, and `mod_proxy_http`, are present and enabled (in other words, are *not* commented out):

```
LoadModule proxy_balancer_module "VFWS-INSTALL/httpd-
    2.2
    2.4
    /modules/mod_proxy_balancer.so"
LoadModule proxy_module "VFWS-INSTALL/httpd-
    2.2
    2.4
    /modules/mod_proxy.so"
LoadModule proxy_http_module "VFWS-INSTALL/httpd-
    2.2
    2.4
    /modules/mod_proxy_http.so"
```

where `VFWS-INSTALL` refers to the directory in which you installed Pivotal Web Server. If they are not in the file, add them in the same location as the other `LoadModule` directives.

Following our example, the directive configurations would be:

```
LoadModule proxy_balancer_module "/opt/pivotal/webserver/httpd-
    2.2
    2.4
    /modules/mod_proxy_balancer.so"
LoadModule proxy_module "/opt/pivotal/webserver/httpd-
    2.2
    2.4
    /modules/mod_proxy.so"
LoadModule proxy_http_module "/opt/pivotal/webserver/httpd-
    2.2
    2.4
    /modules/mod_proxy_http.so"
```

The Pivotal Web Server configuration file is located in the `conf` directory of your Pivotal Web Server instance (`/opt/pivotal/webserver/lb-server/conf` in our example).

3. In the same `httpd.conf` file, add the proxy configuration.

Use the `<Proxy>` element to specify the list of tc Runtime instances and the method of load balancing you want to use. Then use the `ProxyPass` and `ProxyPassReverse` directives to specify the URLs that will use this proxy and load-balancing (both for requests and responses.) For example:

```
<Proxy balancer://my-balancer>
    BalancerMember http://192.168.0.203:8081 route=instanceOne loadfactor=1
    BalancerMember http://192.168.0.203:8082 route=instanceTwo loadfactor=1
    ProxySet lbmethod=byrequests
</Proxy>
```

```
ProxyPass /my-app balancer://my-balancer/my-app
ProxyPassReverse /my-app http://192.168.0.203:8081/my-app
ProxyPassReverse /my-app http://192.168.0.203:8082/my-app
```

In the preceding example:

- The `balancer` parameter of the `<Proxy>` element specifies a unique identifier for this load balancer configuration.
 - Each `tc Runtime` instance that is serviced by this load balancer must have its own `BalancerMember`; the first parameter of this directive specifies the full IP address (including port number) of the `tc Runtime` instance.
 - The `route` parameter contains session ID information. You later use the value of this parameter in the `tc Runtime` configuration file to configure sticky sessions; for now, just ensure that the values are unique for each `BalancerMember`.
 - The `loadfactor` parameter specifies how much load a particular member carries. If you want each member to carry the same load, set the numbers equal to each other (as in the example above). If, however, you want one member to work three times harder than the other, set the load factors to 3 and 1.
 - Use the `lbmethod` parameter of the `ProxySet` directive to specify the load balancing algorithm. The possible values are as follows:
 - `byrequests`: performs weighted request counting. This is the default value.
 - `bytraffic`: performs weighted traffic byte count balancing.
 - `bybusyness`: performs pending request balancing.
 - Use the `ProxyPass` and `ProxyPassReverse` to specify the context URLs of the application that will be routed to the `tc Runtime` instances that you have configured in the load balancing scheme. `ProxyPass` specifies that when the Pivotal Web Server instance receives a request at the `/my-app` URL, it routes the request to the load balancer that will in turn route it to the `tc Runtime` instance. `ProxyPassReverse` does the reverse: when the `tc Runtime` instance sends a response to a user who is using `/my-app`, the response appears to come from the Pivotal Web Server instance, and not the `tc Runtime` instance. Thus the details of the `tc Runtime` instance are *hidden* from the user.
- 4. Optional.** If you want to enable the balancer manager Web application to watch the load balancing activity and control the behavior, add the following to the `httpd.conf` configuration file of your Pivotal Web Server instance:

```
<Location /balancer-manager>
  SetHandler balancer-manager
  Order Deny,Allow
  Deny from all
  # BE VERY RESTRICTIVE with YOUR ALLOW STATEMENT
  Allow from 127.0.0.1
</Location>
```

- 5. Optional.** If you want to enable sticky sessions, follow these steps:

- a.** In the `httpd.conf` file of the Pivotal Web Server instance, update the `ProxySet` directive of the `<Proxy>` element you configured in a preceding *step* by adding the `stickysession=JSESSIONID|jsessionid` parameter. This parameter configures the cookie/path that will be used for stickiness. For example (update shown in bold):

```
<Proxy balancer://my-balancer>
  BalancerMember http://192.168.0.203:8081 route=instanceOne
  loadfactor=1
  BalancerMember http://192.168.0.203:8082 route=instanceTwo
  loadfactor=1
  ProxySet lbmethod=byrequests stickysession=JSESSIONID|jsessionid
</Proxy>
```

- b.** Go to the computer on which Pivotal `tc Server` is running and update the `server.xml` configuration file of both `tc Runtime` instances by adding the `jvmRoute=value` attribute to the `Catalina <Engine>` element. Set the value of this attribute equal to the value you specified (in a preceding

step) for the `route` parameter of the `BalancerMember` directive in the Pivotal Web Server `httpd.conf` file that describes the `tc Runtime` instance.

Following our example, the updated `<Engine>` entry for the `instanceOne` `tc Runtime` instance (that uses port `8081`) would be as follows (new attribute in bold):

```
<Engine name="Catalina" defaultHost="localhost" jvmRoute="instanceOne">
```

If you configure sticky sessions, Pivotal recommends that you also configure session replication for the `tc Runtime` instances. For details, see the section titled *Enabling Clustering for High Availability* in the Pivotal `tc Server` documentation.

6. Start the Pivotal Web Server instance. Following our example:

```
prompt# cd /opt/pivotal/webserver/lb-server
prompt# bin/httpdctl start
```

7. Start (or restart) the two `tc Runtime` instances for the configuration changes to take effect. Following our example:

```
prompt$ cd /var/opt/pivotal/pivotal-tc-server-standard
prompt$ ./tcruntime-ctl.sh instanceOne restart
prompt$ ./tcruntime-ctl.sh instanceTwo restart
```

You have now configured load balancing for the two `tc Runtime` instance using the front-end Pivotal Web Server.

What to do next

- For full reference documentation on the directives described in step 3, along with additional parameters you can use, see *Apache Module mod_proxy* on the Apache Software Foundation Web site.
- Ensure that you can access your application through the Pivotal Web Server instance, which in turn routes the request to one of the `tc Runtime` instances. Do this by invoking your application in a browser, but specify the Pivotal Web Server instance rather than the `tc Runtime` instance. For example, if the URL to access the Pivotal Web Server is `http://www.myhost.com`, invoke the `/my-app` application using the following URL in your browser:

```
http://www.myhost.com/my-app
```

If you see your application, then you have correctly set up the Pivotal Web Server instance to route requests to the `/my-app` application to one of the two `tc Runtime` instances. The Pivotal Web Server instance will also balance the load between the two instances.

- If you enabled the balancer manager Web application, use it to watch and control load-balancing activity. Access the balancer manager application by navigating to the following URL in your browser:

```
http://localhost:port/balancer-manager
```

where `port` is the port number of the Pivotal Web Server instance (`80` by default.) For security, the balancer manager configuration allows access only to users who navigate to the application using a browser installed on the same computer on which the Pivotal Web Server instance is actually running.

Configure SSL Between Pivotal Web Server and Pivotal tc Server

For additional security, it is often desirable to configure SSL between a Pivotal Web Server instance and one or more `tc Runtime` instances, although it's not required. (*tc Runtime* is the runtime component of Pivotal `tc Server`.)

SSL certificates are frequently used to confirm the identity of a server before consuming its services and to secure communications with the server. Typically, if you use a Pivotal Web server instance to load balance requests to one or more `tc Runtime` instances, the SSL encryption and certificate authentication is terminated at the Web Server instance. Communication between the Web Server and `tc Runtime` instances is then trusted and in clear text.

However, there are organizational security policies and B2B scenarios that might mandate secure communication between the Pivotal Web Server and tc Runtime instances. Furthermore, it might be important to restrict access to the tc Runtime instances to known instances of Pivotal Web Server.

This section provides details for configuring SSL communication and client certificate authentication between Pivotal Web Server and tc Server. The high-level steps are as follows, with detailed information about each step in its own sub-section:

1. *Configure tc Runtime Instances to Use SSL*
2. *Configure the Pivotal Web Server instance to Use SSL*
3. *Update the Web Server Configuration for HTTPS Connections to tc Runtime Instances*
4. *Restrict Communication With tc Runtime Instances to Known Clients*
5. *Configure Pivotal Web Server to Authenticate Using a Specific Client Certificate*

Important: It is assumed that you have already installed Pivotal Web Server and Pivotal tc Server, created instances, and set up unsecured load balancing between them. If you have not already done this, see *Configure Load Balancing Between Pivotal Web Server and Pivotal tc Server*.

Configure tc Runtime Instances to Use SSL

Pivotal recommends that you configure a tc Runtime instance to use SSL by specifying the `bio-ssl` template when you create or modify an instance; this template adds the correct configuration to the `conf/server.xml` file and automatically generates a keystore based on your inputs. You specify the `bio-ssl` template when you create a new tc Runtime instance using the `tcruntime-instance` command. Additionally, as of version 2.8 of Pivotal tc Server, you can also apply the template to an existing instance.

The following example shows how to create a new tc Runtime instance that uses the `bio-ssl` template:

```
prompt$ ./tcruntime-instance.sh create instanceOne -t bio-ssl -i /var/opt/pivotal/pivotal-tc-server-standard
```

In the preceding example, the tc Runtime instance will be located in the `/var/opt/pivotal/pivotal-tc-server-standard` directory and will use default values when creating the keystore. If you want to customize the keystore, use the `--interactive` option and the command will prompt you for specific information:

```
prompt$ ./tcruntime-instance.sh create instanceOne -t bio-ssl -i /var/opt/pivotal/pivotal-tc-server-standard --interactive
```

The following example shows how to apply the `bio-ssl` template to an existing tc Runtime instance called `instanceTwo`:

```
prompt$ ./tcruntime-instance.sh apply-template instanceOne -t bio-ssl -i /var/opt/pivotal/pivotal-tc-server-standard
```

Note: The `apply-template` option of `tcruntime-instance` is available as of version 2.8 of Pivotal tc Server.

To invoke an application deployed to the tc Runtime instance using HTTPS, specify the HTTPS port. The default HTTPS port is 8443, although you might have configured a different port for your particular instance. For example:

```
https://host:8443/my-app
```

See *Create and Modify a tc Runtime Instance* in the *Getting Started with Pivotal tc Server* guide in this Documentation Center for details.

If you chose not to use the `bio-ssl` template, you can create your own keystore using the `keytool` command, as shown in the following example:

```
prompt$ keytool -genkey -alias tomcat -keyalg RSA -keystore CATALINA_BASE/conf/tomcat.keystore
```

In the preceding example, `CATALINA_BASE` refers to the instance directory, such as `/var/opt/pivotal/pivotal-tc-server-standard/instanceOne`.

Update the appropriate `<Connector />` element in the instance's `conf/server.xml` file by adding the `keyAlias`, `keystoreFile`, and `keystorePass` attributes, setting the values to those you specified when you created the keystore using `keytool` as shown above. For example:

```
<Connector SSLEnabled="true"
  acceptCount="100"
  connectionTimeout="20000"
  executor="tomcatThreadPool"
  keyAlias="tomcat"
  keystoreFile="${catalina.base}/conf/tomcat.keystore"
  keystorePass="changeme"
  maxKeepAliveRequests="15"
  port="8443"
  protocol="org.apache.coyote.http11.Http11Protocol"
  redirectPort="8443"
  scheme="https"
  secure="true"/>
```

Configure the Pivotal Web Server Instance to Use SSL

The easiest way to do configure SSL for a Web Server instance is to use the `newserver` interactive command to create a new instance and specify that you want to enable SSL for the instance. The command performs configuration tasks and creates a private key. See [Create Pivotal Web Server Instances](#).

If you want to enable SSL for an *existing* Web Server instance, you can uncomment the `Include conf/extras/httpd-ssl.conf` directive in the main `conf/httpd.conf` file and then follow directions in the `conf/extra/httpd-ssl.conf` file. See [Using the Sample Configuration Files to Enable Features and Modify Configuration](#) for details.

Restrict Communication With tc Runtime Instances to Known Clients

This section describes how to specify that the tc Runtime instances require a valid certificate from a client before it accepts a connection.

Procedure

1. Log in to the computer on which you installed tc Server and open a terminal window.
2. Create a certificate authority file. The following examples show how to create the CA file `ca.crt` by using the `openssl` command:

```
prompt$ openssl genrsa -out ca.key 1024
prompt$ openssl req -new -key ca.key -out ca.csr
prompt$ openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt
```

3. Generate a JKS formatted certificate authority file. The following example shows how to use the `keytool` command to create the file `cacerts.jks` from the CA file `ca.crt` you created in the preceding step:

```
prompt$ keytool -importcert -keystore cacerts.jks -storepass changeme -alias
my_ca -file ca.crt
```

4. Copy the `cacerts.jks` file to the `CATALINA_BASE/conf` directory of each tc Runtime instance.
5. Update the appropriate `<Connector />` element in each instance's `conf/server.xml` file by adding the `clientAuth` and `truststoreFile` attributes, setting their values as shown in the example:

```
<Connector SSLEnabled="true"
  clientAuth="true"
  truststoreFile="${catalina.base}/conf/cacerts.jks"
  acceptCount="100"
  connectionTimeout="20000"
  executor="tomcatThreadPool"
  keyAlias="tomcat"
  keystoreFile="${catalina.base}/conf/tomcat.keystore"
  keystorePass="changeme"
```

```
maxKeepAliveRequests="15"
port="8443"
protocol="org.apache.coyote.http11.Http11Protocol"
redirectPort="8443"
scheme="https"
secure="true"/>
```

- Restart each tc Runtime instance for the changes to take effect:

```
prompt$ cd /var/opt/pivotal/pivotal-tc-server-standara/instanceOne/bin
prompt$ ./tcruntime-ctl.sh restart
```

- Test that you have configured each tc Runtime instance correctly by navigating to an application deployed on the instance using your browser.

The tc Runtime instance should deny you access because your browser does not have the required client certificate configured.

Update the Web Server Configuration for HTTPS Connections to tc Runtime Instances

As specified earlier, it is assumed that you have already configured your Pivotal Web Server instance for unsecured load balancing between two or more tc Runtime instances. If you have not already done this, see [Configure Load Balancing Between Pivotal Web Server and Pivotal tc Server](#).

Update the Web Server configuration to communicate with the tc Runtime instances securely by editing the `conf/http.conf` file in the Web Server instance directory (such as `/opt/pivotal/webserver/lb-server`) and changing the tc Runtime URLs so they use HTTPS and specify the HTTPS port. Following the example from the load balancing section, if you specified that the HTTPS ports for `instanceOne` and `instanceTwo` were 8443 and 8553, respectively, the updated file would look like this:

```
<Proxy balancer://my-balancer>
  BalancerMember https://192.168.0.203:8443 route=instanceOne loadfactor=1
  BalancerMember https://192.168.0.203:8553 route=instanceTwo loadfactor=1
  ProxySet lbmethod=byrequests
</Proxy>

ProxyPass /my-app balancer://my-balancer/my-app
ProxyPassReverse /my-app https://192.168.0.203:8443/my-app
ProxyPassReverse /my-app https://192.168.0.203:8553/my-app
```

Configure Pivotal Web Server to Authenticate Itself Using a Specific Client Certificate

This section describes how to configure the Web Server instance to authenticate itself using the client certificate you created in a previous step and configured for each tc Runtime instance.

- On the computer on which you installed Pivotal Web Server, create a client certificate and key. Use the *same* certificate authority file (called `ca.crt` in the example) that you created in a *previous step*.

The following example shows how to do this using the `openssl` command:

```
prompt$ openssl genrsa -out client.key 1024
prompt$ openssl req -new -key client.key -out client.csr -config your-
openssl.cnf-file
prompt$ openssl x509 -req -days 365 -CA ca.crt -CAkey ca.key -CAcreateserial
-in client.csr -out client.crt
```

In the preceding example, the newly generated client key is called `client.key` and the client certificate file is called `client.crt`. Replace `your-openssl.cnf-file` with the full path name of the `openssl.cnf` file on your computer, such as `/etc/pki/tls/openssl.cnf`.

- Concatenate the generated client key and client certificate files into a single file. In the following example, the new file is called `client.crtkey`:

```
prompt$ cat client.crt client.key > client.crtkey
```

3. Copy the generated `client.crtkey` file to the `ssl` directory of the Web Server instance directory. For example:

```
prompt$ cp client.crtkey /opt/pivotal/webserver/lb-server/ssl
```

4. Configure the `mod_ssl` module of the Web Server instance to use SSL for the proxy engine and to use the generated client certificate and key file by adding the following directives to the file that contains the SSL configuration, such as `conf/extra/httpd-ssl.conf`:

```
SSLProxyMachineCertificateFile "ssl/client.crtkey"
SSLProxyEngine on
```

5. Restart the Pivotal Web Server instance for the configuration changes to take effect. For example:

```
prompt# cd /opt/pivotal/webserver/lb-server
prompt# bin/httpdctl restart
```

6. Test that everything is working correctly by accessing your application through the Pivotal Web Server host or IP address and HTTPS port. For example, if the Web Server IP address is `192.11.22.33` and configured an HTTPS port of `8663`:

```
https://192.11.22.33:8663/my-app
```

Because you have configured your Web Server instance with the client certificate required by the `tc Runtime` instances, you will see your application, and yet all communication from your browser to the `tc Runtime` instance is using SSL.

Configure BMX for Monitoring Pivotal Web Server Instances

As of version 5.1, all new Pivotal Web Server instances are configured with BMX by default.

BMX is an Apache HTTPD framework that provides internal runtime information (performance metrics, status, configuration, and current capacity) to monitoring applications such as Pivotal Hyperic. In turn, these types of applications monitor the health of Pivotal Web Server instances by running BMX queries to gather metrics and configuration information.

New Pivotal Web Server instances have the following default BMX configuration:

- Three main BMX modules (`mod_bmx`, `mod_bmx_status`, and `mod_bmx_vhost`) are all enabled. Together, these modules provide overall runtime statistics of the Web Server instance, as well as the virtual hosts running within the instance.
- Access is allowed only to processes running on `http://localhost` (IP address `127.0.0.1`), or in other words, only to monitoring applications running on the same computer as the Web Server instance.
- Access requires no authentication.
- BMX access is enabled for all virtual hosts defined for the Web Server instance.

The default BMX configuration for Pivotal Web Server instances make them immediately available for monitoring by monitoring applications.

The BMX-related modules are loaded into the Web Server instance using appropriate `LoadModule` directives in the `conf/httpd.conf` configuration file. Additional BMX configuration is in the `conf/extra/httpd-info.conf` file, which the main `conf/httpd.conf` file includes using the `Include conf/extra/httpd-info.conf` directive.

Procedure

1. To disable BMX access to your Pivotal Web Server instance, comment out the appropriate `LoadModule` directives in the `conf/httpd.conf` configuration file for your instance as shown:

```
#LoadModule bmx_module c:/opt/pivotal/webserver/httpd-
    2.2
    2.4
    /modules/mod_bmx.so
#LoadModule bmx_status_module c:/opt/pivotal/webserver/httpd-
```

```

                2.2
                2.4
                /modules/mod_bmx_status.so
#LoadModule bmx_vhost_module c:/opt/pivotal/webserver/httpd-
                2.2
                2.4
                /modules/mod_bmx_vhost.so

```

2. To allow BMX access to processes running on hosts other than the `localhost`, edit the `<Location /bmx>` directive in the `conf/extra/httpd-info.conf` file and add the IP address or fully qualified domain name to the `Allow from` directive. For example, to allow `myhost.com` access in addition to `localhost`:

```

<Location /bmx>
    SetHandler bmx-handler
    Order Deny,Allow
    Deny from all
    Allow from 127.0.0.1 myhost.com
</Location>

```

3. To restrict BMX access to a particular virtual host, put the `<Location /bmx>` directive inside the appropriate `<VirtualHost>` directive. For example:

```

<VirtualHost 10.1.2.3:80>
    DocumentRoot "/opt/pivotal/webserver/myserver/myhost.com/htdocs"
    ServerName status.myhost.com
    ...
    <Location /bmx>
        SetHandler bmx-handler
        Order Deny,Allow
        Deny from all
        Allow from 127.0.0.1 myhost.com
    </Location>
</VirtualHost>

```

What to do next

- Restart the Pivotal Web Server instance for the configuration changes to take effect. For example, on Unix:

```

prompt# cd /opt/pivotal/webserver/myserver
prompt# bin/httpdctl restart

```

Metrics

This section lists the metrics reported by the Hyperic plugin for Pivotal Web Server.

Pivotal Web Server Server Metrics

This section lists the metric available for a server.

Table 9: Metric Definitions

Metric	Alias	Units	Category	Default On	Default Interval
Availability	Availability	percentage	Availability	true	5 min
Server Uptime	ServerUptime~Seconds	none	Availability	false	5 min
Busy Workers	BusyWorkers	none	Utilization	true	5 min
Idle Workers	IdleWorkers	none	Utilization	true	5 min

Metric	Alias	Units	Category	Default On	Default Interval
Bytes Served Per Second	KilobytesPerSec	KB	Throughput	false	5 min
Bytes Served Per Request	KilobytesPerReq	KB	Throughput	false	5 min
Requests Served Per Second	ReqPerSec	none	Throughput	true	5 min
Requests Served	TotalAccesses	none	Throughput	false	10 min
Requests Served per Minute	TotalAccesses~1m	none	Throughput	true	10 min
Bytes Served	TotalTraffic~Kilobytes	KB	Throughput	false	10 min
Bytes Served per Minute	TotalTraffic~Kilobytes1m	KB	Throughput	true	10 min
Parent Server Generation	ParentServer~Generation	none	Throughput		

Pivotal Web Server Virtual Host Metrics

This section lists the metrics available for a virtual host.

Table 10: Metric Definitions

Metric	Alias	Units	Category	Default On	Default Interval
Availability	Availability	percentage	Availability	true	10 min
Start Elapsed	StartElapsed	mu	Availability	true	10 min
Start Time	StartTime	epoch-millis	Availability	true	10 min
In Bytes GET	InBytesGET	none	Throughput	false	10 min
In Bytes GET per Minue	InBytesGET1m	none	Throughput		10 min
In Bytes HEAD	InBytesHEAD	none	Throughput	false	10 min
In Bytes HEAD per Minute	InBytesHEAD1m	none	Throughput	true	10 min
In Bytes POST	InBytesPOST	none	Throughput	false	10 min
In Bytes POST per Minute	InBytesPOST1m	none	Throughput	true	10 min
In Bytes PUT	InBytesPUT	none	Throughput	false	10 min
In Bytes PUT per Minute	InBytesPUT1m	none	Throughput	true	10 min

Metric	Alias	Units	Category	Default On	Default Interval
In Requests GET	InRequestsGET	none	Throughput	false	10 min
In Requests GET per Minute	InRequests~GET1m	none	Throughput	true	10 min
In Requests HEAD	InRequests~HEAD	none	Throughput	false	10 min
In Requests HEAD per Minute	InRequests~HEAD1m	none	Throughput	true	10 min
In Requests POST	InRequests~POST	none	Throughput	false	10 min
In Requests POST per Minute	InRequests~POST1m	none	Throughput	true	10 min
In Requests PUT	InRequestsPUT	none	Throughput	false	10 min
In Requests PUT per Minute	InRequests~PUT1m	none	Throughput	true	10 min
Out Bytes 200	OutBytes200	none	Throughput	false	10 min
Out Bytes 200 per Minute	OutBytes2001m	none	Throughput	true	10 min
Out Bytes 301	OutBytes301	none	Throughput	false	10 min
Out Bytes 301 per Minute	OutBytes3011m	none	Throughput	true	10 min
Out Bytes 302	OutBytes302	none	Throughput	false	10 min
Out Bytes 302 per Minute	OutBytes3021m	none	Throughput	true	10 min
Out Bytes 403	OutBytes403	none	Throughput	false	10 min
Out Bytes 403 per Minute	OutBytes4031m	none	Throughput	true	10 min
Out Bytes 404	OutBytes404	none	Throughput	false	10 min
Out Bytes 404 per Minute	OutBytes4041m	none	Throughput	true	10 min
Out Bytes 500	OutBytes500	none	Throughput	false	10 min
Out Bytes 500 per Minute	OutBytes5001m	none	Throughput	true	10 min
Out Responses 200	OutResponses200	none	Throughput	false	10 min
Out Responses 200 per Minute	OutResponses2001m	none	Throughput	true	10 min

Metric	Alias	Units	Category	Default On	Default Interval
Out Responses 301	OutResponses301	none	Throughput	false	10 min
Out Responses 301 per Minute	OutResponses301m	none	Throughput	true	10 min
Out Responses 302	OutResponses302	none	Throughput	false	10 min
Out Responses 302 per Minute	OutResponses302m	none	Throughput	true	10 min
Out Responses 401	OutResponses401	none	Throughput	false	10 min
Out Responses 401 per Minute	OutResponses401m	none	Throughput	true	10 min
Out Responses 403	OutResponses403	none	Throughput	false	10 min
Out Responses 403 per Minute	OutResponses403m	none	Throughput	true	10 min
Out Responses 404	OutResponses404	none	Throughput	false	10 min
Out Responses 404 Per Minute	OutResponses404m	none	Throughput	true	10 min
Out Responses 500	OutResponses500	none	Throughput	false	10 min
Out Responses 500 per Minute	OutResponses500m	none	Throughput	true	10 min
In Low Bytes	InLowBytes	none	Throughput	false	10 min
In Low Bytes per Minute	InLowBytes1m	none	Throughput	true	10 min
In Requests	InRequests	none	Throughput	false	10 min
In Requests per Minute	InRequests1m	none	Throughput	true	10 min
Out Responses	OutResponses	none	Throughput	false	10 min
Out Responses per Minute	OutResponses1m	none	Throughput	true	10 min

Security Information

Pivotal is committed to providing products and solutions that allow you to assess the security of your information, secure your information infrastructure, protect your sensitive information, and manage security information and events to assure effectiveness and regulatory compliance. As part of this commitment, the following Pivotal Web Server-specific security information is provided to help you secure your environment:

- *External Ports*
- *Resources That Must Be Protected*
- *Log File Locations*
- *User Accounts Created at Installation*
- *Obtaining and Installing Latest Version of Product*

External Interfaces, Ports, and Services

A Pivotal Web Server instance uses TCP/IP ports to receive incoming requests and send outgoing responses. Different protocols (such as HTTP or HTTPS) listen on different ports. You can change these port numbers when you create the Web Server instance using the `newserver` script, but these are the default values:

- HTTP: 80
- HTTPS: 443

If you have already created the Web Server instance, you can change its HTTP listen port by updating the `Listen 90 http` directive in the `INSTANCE-DIR/conf/httpd.conf` file, where `INSTANCE-DIR` refers to the directory in which the Web Server instance is located, such as `/opt/pivotal/webserver/myserver`. To update the HTTPS port, update the `Listen 443 https` directive in the `INSTANCE-DIR/conf/extra/httpd-ssl.conf` file.

Pivotal Web Server does not have any external interfaces or services that need to be enabled or opened.

Resources That Must Be Protected

The following Pivotal Web Server configuration files should be readable and writable *only* by the `root` (Unix) or `Administrator` (Windows) user:

- `conf/httpd.conf`
- `conf/userfile`
- All files in the `ssl` directory (if you have enabled SSL for the instance)
- `extra/conf/httpd-ssl.conf` (if you have enabled SSL for the instance)
- Any other `conf/httpd-XX.conf` file that you have for which there is an uncommented `Include` in the main `conf/httpd.conf` configuration file.

These configuration files are specific to a Web Server instance and are stored in the `INSTANCE-DIR` directory, where `INSTANCE-DIR` refers to the directory in which the Web Server instance is located, such as `/opt/pivotal/webserver/myserver`.

Log File Locations

The most important log files for a Pivotal Web Server instance are as follows:

- `error_log`: Contains errors and diagnostic information that occurred while the Web Server instance was serving requests.
- `access_log`: Contains information about all Web Server requests.

- `ssl_request_log`: Applies only if you enabled SSL. Contains information about requests that came over HTTPS.

These log files are specific to a Web Server instance and are stored by default in the `INSTANCE-DIR/logs` directory, where `INSTANCE-DIR` refers to the directory in which the Web Server instance is located, such as `/opt/pivotal/webserver/myserver`.

The preceding log files should be readable and writable *only* by the `root` (Unix) or Administrator (Windows) user.

The `logs` directory also contains other log files associated with BMX and the Pivotal License server.

User Accounts Created at Installation

If you install Pivotal Web Server on Red Hat Enterprise Linux (RHEL) using the RPM, a user with the following characteristics is automatically created:

- ID: `pwshttpd`
- Group: `pwshttpd`
- Non-interactive, which means that you cannot directly log in to the RHEL computer as this user. Rather, you must log in as `root` or user with appropriate `sudo` privileges and `su - pwshttpd`.

When installing from RPM on RHEL, the installation directory will be owned by the `root` user, with group `root`.

When installing Web Server on Windows or Unix from a self-extracting `*.zip` file, a user account is *not* automatically created for you. Rather, you should install as `root` on Unix and Administrator on Windows.

Obtaining and Installing Security Updates

Pivotal Web Server an HTTPD server based on open-source Apache HTTPD. Pivotal Web Server includes a particular version of Apache HTTP Server, such as `httpd- 2.2.29.0 2.4.10` . New versions of Pivotal Web Server typically include an updated version of Apache HTTPD, some of which might fix important security vulnerabilities. To install these security updates, you install the new version of Pivotal Web Server and then upgrade your existing instances.

To download the latest `*.zip` distributions of the Pivotal Web Server, go to the [Pivotal Web Server product page](#), and click **Downloads**.

When using RPMs on RHEL, use the `yum upgrade` command to upgrade to the latest Pivotal Web Server version.

See [Upgrading Pivotal Web Server](#) for details.

Additional Documentation

The documentation in this guide provides information about what Pivotal Web Server contains; how to install it; and how to create, start, and stop instances. The Pivotal Web Server documentation does not, however, provide details about configuring and using the core Apache HTTP component; for that you must go elsewhere, such as the Apache documentation.

- Apache HTTP Server 2.2 2.4 <http://httpd.apache.org/docs/2.2/> <http://httpd.apache.org/docs/2.4/>
- OpenSSL <http://www.openssl.org/docs/>
- ASF Bugzilla page (search for known bugs in Apache HTTP Server) <https://issues.apache.org/bugzilla/>
- Searchable archive of Apache HTTP Users mail list <http://marc.info/?l=apache-httpd-users&r=1&w=2>