vFabric Web Server Installation and Configuration

VMware vFabric Web Server 5.0

VMware vFabric Cloud Application Platform 5.0

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see http://www.vmware.com/support/pubs.

EN-000660-00

vmware

You can find the most up-to-date technical documentation on the VMware Web site at: https://www.vmware.com/support/.

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to: docfeedback@vmware.com

VMware, Inc., 3401 Hillview Avenue, Palo Alto, CA 94304

www.vmware.com

Table of Contents

1. About vFabric Web Server Installation and Configuration

VMware vFabric Web Server Installation and Configuration describes product concepts and product-specific configuration tasks for VMware® vFabric™ Web Server. Fully compatible with Apache Web Server, vFabric Web Server is a dynamic load-balancing service that is available exclusively with VMware® vFabric™ Cloud Application Platform. Read this documentation for an overview of vFabric Web Server features, installation instructions, and information on how to configure functionality that is specific to vFabric Web Server.

Intended Audience

vFabric Web Server Installation and Configuration is intended for experienced Windows and Linux developers and system administrators who want to install a Web Server on a virtual machine to serve static Web content, act as a proxy, or load balance between application servers such as VMware® vFabric™ tc Server.

2. Overview of vFabric Web Server

vFabric Web Server is the Web server and load-balancing component of VMware vFabric Cloud Application Platform (vFabric Platform). vFabric Web Server 5.0 is based on Apache HTTP Server version 2.2.

In addition to the standard features of Apache HTTP Server, vFabric Web Server provides the following mission-critical benefits:

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

Subtopics

Complete Packages and Modules in vFabric Web Server 5.0

This section lists the complete contents of vFabric Web Server 5.0.

This document describes features and functionality for vFabric Web Sever 5.0. For general information about new features in Apache HTTP Server 2.2, see the Apache Web site.

- Core Apache HTTP 2.2 Binaries
- Apache mod_ftp 0.9.6
- Apache mod_fcgid 2.3.6 (a FastCGI invoker)
- OpenSSL 0.9.8r
- OpenLDAP 2.3.43
- mod_jk 1.2.31
- mod_bmx (Hyperic plug-in for monitoring support)
- libexpat 2.0.1
- libapr 1.4.3
- libaprutil 1.3.11
- zlib 1.2.3

mod_fcgid Implementation of Connector to FastCGI

The mod_fcgid distributed with vFabric Web Server 5.0 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.

Open SSL 0.9.8r Supports SSLFIPS and SSLInsecureRenegotiation Directives

The mod_ssl directive SSLInsecureRenegotiation supports per-directory/per-server client certificate renegotiation with legacy clients.

Users are strongly cautioned to update all clients to use only the new TLS Renegotiation mechanism.

The OpenSSL 0.9.8r build is vendor-affirmed to comply with OpenSSL FIPS Security Policy document v1.2. In order to use the FIPS 140-2 validated mode of operation (which relies on NIST-approved, but slower algorithms) for mod_ssl cryptography, simply add the new 'SSLFIPS on' directive. A log entry indicates whether FIPS mode is enabled. A warning that 512-byte temporary key generation is skipped is logged at startup. This behavior is correct (512 byte keys are disallowed for FIPS use). See Apache Module mod_ssl .

Supported Platforms

TCP/IPv6 is required on all platforms. Modern platforms have this support installed already. It is not necessary to configure IPv6 adapters; simply install the IPv6 socket support drivers and associated libraries. Vendor patch sets for compatibility with the current release of vendor Java are recommended for all vFabric Web Server installations, because the operating system flaws that affect sockets and multi-threaded services for Java similarly affect vFabric Web Server and its support libraries.

Table 2.1. vFabric Web Server 5.0 Supported Platforms

Operating System	Processor	Package Format
Red Hat Enterprise Linux (RHEL) 5	x86 (32- and 64-bit)	RPM and self-extracting ZIP (`.zip.sfx`)
Ubuntu 10.04	x86 (64-bit)	Self-extracting ZIP (`.zip.sfx`)
Windows 2003	x86 (32-bit)	Self-extracting ZIP (`.zip.exe`)
Windows 2008	x86 (32- and 64-bit)	Self-extracting ZIP (`.zip.exe`)

3. Installing vFabric Web Server

vFabric Web Server is available only as part of vFabric Cloud Application Platform, which in turn runs only on vSphere. This means that you install vFabric Web Server on virtual machines that run on the vCenter Server. These VMs can run Red Hat Enterprise Linux, Ubuntu Linux, or Windows operating systems. Each type of operating system requires its own installation procedure.

Subtopics

RHEL: Install vFabric Web Server from the VMware Package Repository

When installing vFabric Web Server on a Red Hat Linux Enterprise (RHEL) virtual machine, it is recommended that you use `yum` from the VMware package repository.

Prerequisites

- Create a virtual machine on your vCenter Server and install RHEL on it. See Supported Platforms for the list of supported platforms for vFabric Web Server.
- Log in to the RHEL VM as the `root` user and start a terminal.
- Install the `vfabric-5-repo` RPM using the `rpm` command. This RPM makes it easy for you to browse the vFabric RPMs available from the VMware repository:

  ```
  prompt# rpm -Uvh http://repo.vmware.com/pub/rhel5/vfabric/5/vfabric-5-repo-5-2.noarch.rpm
  ```

  Use the `yum search vfabric` command to view the list of vFabric components that you can install from the VMware repository. The vFabric Web Server RPM is called `vfabric-web-server`.

Procedure

1. From the RHEL VM on which you will install vFabric Web Server, log in as the `root` user.
2. Execute the following `yum` command:

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

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

   The `yum` command automatically chooses the appropriate RPM package based on your architecture (32- or 64-bit). You can also install the RPM with a particular architecture by specifying the appropriate suffix. For example, to install the 64-bit RPM:

   ```
   prompt# yum install vfabric-web-server.x86_64
   ```

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

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

   The `yum` command:

- Installs vFabric Web Server into the `/opt/vmware/vfabric-web-server` directory.
- Sets the owner of all directories and files under `/opt/vmware/vfabric-web-server` to `root`.

What to do next

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

Linux: Install vFabric Web Server from a Self-Extracting ZIP

You can install vFabric Web Server on Linux VMs with a self-extracting ZIP file that you download from the VMware 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

- Create a virtual machine on your vCenter Server and install Linux on it. See Supported Platforms for the list of supported platforms for vFabric Web Server.
- 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 VM, and that it is at least version 5.8. Additionally, you cannot run the Perl script (described in the procedure) in multibyte character encoding such as UTF8; this means you must unset the LANG or LC_*** variables to remove any UTF8 or other encodings.

Procedure

1. Log in as the `root` user on to the Linux VM on which you want to install vFabric Web Server.
2. Create the directory in which you will install vFabric Web Server. For example, on Linux:

   ```
   prompt# mkdir /opt/vmware
   ```

3. Download the appropriate vFabric Web Server self-extracting ZIP from the VMware Support and Downloads Web site 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 vFabric Web Server on a 64-bit Linux platform is `vfabric-web-server-5.0.0-httpd-2.2.19-x86_64-linux-glibc2.zip.sfx`.

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

   ```
   prompt# cd /opt/vmware
   ```

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

   ```
   prompt# chmod 755 vfabric-web-server-5.0.0-httpd-2.2.19-x86_64-linux-glibc2.zip.sfx
   ```

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

   ```
   prompt# ./vfabric-web-server-5.0.0-httpd-2.2.19-x86_64-linux-glibc2.zip.sfx
   ```

   When it completes, the vFabric Web Server files are located in the `vfabric-web-server` subdirectory.

7. Change to the `vfabric-web-server` directory and run the `fixrootpath.pl` Perl script to correct the root paths. For example:

```
prompt# cd vfabric-web-server
prompt# perl fixrootpath.pl
```

What to do next

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

Windows: Install vFabric Web Server from a ZIP File

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

Prerequisites

- Create a virtual machine on your vCenter Server and install Windows on it. See Supported Platforms for the list of supported platforms for vFabric Web Server.

Procedure

1. From the Windows VM on which you want to install vFabric Web Server, log in as the `Administrator` user.
2. Start a command window, then create the directory into which you will install vFabric Web Server. Do not create a directory name that contains spaces. For example:

```
prompt> mkdir c:\opt\vmware
```

Note: Depending on the Windows version, you may not have the required permissions when you start the command window, even if you logged in as the Administrator user. If, when executing the steps in this procedure, you find that some commands fail because of a lack of permissions, start a new command window from the original one using the `runas` command and run the commands from there instead:

```
prompt> runas /user:administrator "cmd.exe /k"
```

3. Download the appropriate vFabric Web Server self-extracting ZIP from the VMware Support and Downloads Web site and place it in the directory you created; for example, `c:\opt\vmware`.

   Be sure to choose the correct architecture (32- or 64-bit). For example, the file to install vFabric Web Server on a 64-bit Windows platform is `vfabric-web-server-5.0.0-x64-windows.zip.exe`.

4. 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 vFabric Web Server files are located in the `vfabric-web-server` subdirectory.

5. From your command window, change to the main vFabric Web Server directory:

```
prompt> cd c:\opt\vmware\vfabric-web-server
```

6. Run the `fixrootpath.vbs` VBScript to correct the root paths; use the `cscript` command to invoke the script. For example:

```
prompt> cscript fixrootpath.vbs
```

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

   For example, to use the 64-bit edition of vFabric Web Server on a Windows 2008 computer, run the following command:

   ```
   prompt> mklink /d httpd-2.2 httpd-2.2.19.0-64
   ```

   On Windows 2003, use the `makelink` command, which is included in the main vFabric Web Server directory (such as `c:\opt\vmware\vfabric-web-server`). For example:

   ```
   prompt> makelink httpd-2.2 httpd-2.2.19.0-64
   ```

What to do next

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

Description of vFabric Web Server Installation

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

To implement this separation, you use the `newserver.pl` command to create a vFabric Web Server *instance* that lives in a subdirectory of the main vFabric 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`.

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`).

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 vFabric Web Server, you see the following files and directories:

- `httpd-2.2/` : Symbolic link to a sibling directory that actually contains the Apache 2.2 binary runtime files.
- `httpd-2.2.`*`version`*`/` : Directory that contains the actual Apache 2.2 binary runtime files. The `2.2.`*`version`* string specifies the Apache HTTP version, such as `2.2.19.0` or `2.2.19.0-64`.
- `licenses/` : EULA and open source license files.
- `newserver.pl` : (Linux) Perl script for creating vFabric Web Server instances.
- `newserver.vbs` : (Windows) VBscript for creating vFabric Web Server instances.
- `fixrootpath.pl` : (Linux) Perl script for fixing root paths; run only once.
- `fixrootpath.vbs` : (Windows) VBscript for fixing root paths; run only once.

After you use `newserver` to create a new vFabric Web Server instance, the command creates a new directory that contain a separately configurable instance of vFabric 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 vFabric 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 vFabric 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 vFabric Web Server in `/opt/vmware/vfabric-web-server` and create an instance called `myserver`:

```
prompt$ cd /opt/vmware/vfabric-web-server/myserver
prompt$ ls
bin  cgi-bin  conf  ftpdocs  htdocs  logs  proxy ssl  var
```

The `conf` directory contains the vFabric 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.

4. Creating and Using vFabric Web Server Instances

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

Subtopics

Description of vFabric Web Server Instances

A vFabric 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 Linux 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 vFabric Web Server instances running at the same time on the same Linux 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 vFabric 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 vFabric Web Server in `/opt/vmware/vfabric-web-server` and create an instance called `myserver`:

```
prompt$ cd /opt/vmware/vfabric-web-server/myserver
prompt$ ls
bin  cgi-bin  conf  ftpdocs  htdocs  logs  proxy ssl  var
```

The `conf` directory contains the vFabric 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 vFabric Web Server Instances

You create a new vFabric 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.pl` : Perl script for Linux operating systems.
- `newserver.vbs`: VBScript file for Windows operation 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 Linux or Windows.

Prerequisites

- Complete the appropriate procedure in Installing vFabric Web Server.

Procedure

1. Log on to your computer as `root` (Linux) or the Administrator user (Windows) and open a terminal (Linux) or command window (Windows).

   Note for Windows: Depending on the Windows version, you may not have the required permissions when you start the command window, even if you logged in as the Administrator user. If some commands in this procedure fail because of a lack of permissions, start a new command window and use the `runas` command to execute commands from the new window instead of the original one:

   ```
   prompt> runas /user:administrator "cmd.exe /k"
   ```

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

   ```
   prompt# cd /opt/vmware/vfabric-web-server
   ```

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 vFabric Web Server instance. On Linux, use the Perl flavor; for example:

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

   On Windows, use the `cscript` command to invoke the VBS flavor:

   ```
   prompt> cscript newserver.vbs --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/vmware/vfabric-web-server/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 vFabric Web Server instance and verify that it is working correctly. See Linux: Start and Stop vFabric Web Server Instances and Windows: Start and Stop vFabric Web Server Instances.
- Complete the procedure in Serve a Sample HTML File from your vFabric Web Server Instance.
- Configure your instance as described in Configuring vFabric Web Server.

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.pl` : Perl script for Linux.
- `newserver.vbs`: VBScript file for Windows.

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

Table 4.1. Options of the newserver Command

Option	Description	Required?
`--server=`*`servername`*	Name of the new vFabric Web Server instance. The value of this option also can be the name of the directory that contains the instance configuration files, and by default is the name of the host.	Yes.
`--rootdir=`*`rootdir`*	Directory that contains the `httpd-2.2.`*`version`* directory, which in turn contains the Apache HTTP binaries. The default value is the current directory.	No.
`--serverdir=`*`serverdir`*	Directory in which you want the new instance directory to be created. The default value is `rootdir`.	No.
`--overlay`	Specifies that, if `serverdir` exists, you want to overwrite the existing files with new ones.	No. If you do not specify this option, and `serverdir` exists, the `newserver` command returns an error and suggests you specify a unique name and directory location for the new instance.
`--httpdver=`*`httpdver`*	Version of the Apache HTTP binaries you want your instance to use. The default value is `2.2`, which is a symbolic link to the actual installed version of the binaries, such as `2.2.19.0-32`.	No.
`--httpddir=`*`httpddir`*	Directory that contains the Apache HTTP binaries. The default value is `rootdir/httpd-`*`httpdver`*, such as `/opt/vmware/vfabric-web-server/httpd-2.2.19.0-32`.	No.
`--sourcedir=`*`sourcedir`*	Name of the directory that contains the template that `newserver` uses to create the new vFabric Web Server instance. The default value is `httpdir/_instance`.	No.
`--quiet`	Specifies that the `newserver` command should use default values for all prompts.	No. If you do not specify this option, `newserver` interactively prompts for all answers.

Table 4.2. newserver Prompts

Prompt	Description
Use threaded 'worker' MPM [y/n]?	Specifies whether the vFabric Web Server instance implements a multi-process, multi-threaded server. By using threads to serve requests, the instance can serve many requests with fewer system resources than a process-based server.
Enable SSL and create a default key [y/n]?	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.
Server hostname (e.g. www.example.com) [myserver]?	Name that the vFabric 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 `--server` option.

Prompt	Description
Administrator email [webmaster@myserver]?	Email address to which vFabric Web Server instances send problems. This address appears on some instance-generated pages, such as error documents.
Port for http:// traffic [80]?	HTTP port to which the vFabric Web Server instance listens. Default value is `80` when running the `newserver` command as the `root` user on Linux, `8080` otherwise.
Port for https:// SSL traffic [443]?	HTTPS port to which the vFabric Web Server instance listens. Default value is `443` when running the `newserver` command as the `root` user on Linux, `8443` otherwise.
Creating a sample conf/userfile, add initial users	Prompts for creating a sample file that contains the list of users and passwords for authentication. Use this file if you later need to authenticate users who use the vFabric Web Server instance. The prompts ask you for the username and their password; enter Return at the username prompt when you are done.
If you previously specified that you want to enable SSL...	The `newserver` 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. 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.) When `newserver` completes, it generates the following SSL files in the `ssl` subdirectory of the instance directory: • *`instancename`*`.key`: Unencrypted private key. The file has a permission code of 0600 for additional security. • *`instancename`*`.pem`: DES 3 encrypted private key. • *`instancename`*`.csr`: Certificate-signing request. Submit this file to the Certificate Authority. • *`instancename`*`.crt`: Self-signed certificate. Replace this certificate with a signed certificate by the CA. **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.

Linux: Start and Stop vFabric Web Server Instances

You start, stop, or restart a vFabric Web Server instance on Linux 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/vwmare/vfabric-web-server/myserver/bin`. Do not use the start script in the `httpd-2.2/bin` sub-directory of the main installation directory.

Prerequisites

- Complete the appropriate procedure in Create vFabric Web Server Instances.

Procedure

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

```
prompt# cd /opt/vmware/vfabric-web-server/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 vFabric 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 stop the instance immediately, even if there are current connections in use:

```
prompt# cd /opt/vmware/vfabric-web-server/myserver/bin
prompt$ ./httpdctl stop
```

To stop the instance gracefully:

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

6. To restart a currently running instance:

```
prompt$ ./httpdctl restart
```

What to do next

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

Windows: Start and Stop vFabric Web Server Instances

You start, stop, or restart a vFabric Web Server instance on Windows by first installing it as Windows service using the `httpdctl.bat` 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:\vmware\vfabric-web-server\myserver\bin`. Do not use the start script in the `httpd-2.2\bin` sub-directory of the main installation directory.

Prerequisites

- Complete the appropriate procedure in Create vFabric Web Server Instances.

Procedure

1. Log in to your Windows computer as the `Administrator` user and start a command window.

Note for Windows: Depending on the Windows version, you may not have the required permissions when you start the command window, even if you logged in as the Administrator user. If some commands in this procedure fail because of

a lack of permissions, start a new command window and use the `runas` command to execute commands from the new window instead of the original one:

```
prompt> runas /user:administrator "cmd.exe /k"
```

2. Change to the `bin` subdirectory of the root directory for the vFabric Web Server instance.

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

   ```
   prompt> cd c:\vmware\vfabric-web-server\myserver\bin
   ```

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

   ```
   prompt> httpdctl.bat install
   ```

   Subsequently use the Windows Services console to start, stop, and restart the vFabric Web Server instance. The name of the instance in the Windows Services console is `vFabric httpd` *`servername`*`.` Also use the Windows Services control panel to configure whether the service starts automatically when Windows starts, and so on.

4. To test that the vFabric 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 `8080.`

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

   ```
   http://localhost:8080
   ```

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

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

   ```
   prompt> cd c:\vmware\vfabric-web-server\myserver\bin
   prompt> httpdctl.bat uninstall
   ```

What to do next

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

Serve a Sample HTML File from Your vFabric Web Server Instance

After you install vFabric 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 vFabric Web Server instance. See Create vFabric 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 vFabric 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, vFabric 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/vmware/vfabric-web-server/myserver/conf/httpd.conf`. The `DocumentRoot` directive looks like the following:

```
DocumentRoot "/opt/vmware/vfabric-web-server/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/vmware/vfabric-web-server/myserver/htdocs
   ```

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

   For example, if your browser is running on the same computer as vFabric 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/vmware/vfabric-web-server/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 vFabric Web Server instances to take advantage features such as load balancing, virtual hosts, and SSL. See Configuring vFabric Web Server Instances.

5. Configuring vFabric Web Server Instances

The default configuration of a newly created vFabric 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 Server instances. This chapter provides some information to get you started.

For complete documentation on how to configure vFabric Web Server instances, see Apache HTTP Server Version 2.2 Documentation. Because vFabric Web Server is based on Apache HTTP server, the general configuration documentation on the Apache Web site applies to vFabric Web Server as well.

Subtopics

Using Sample Configuration Files to Enable Features and Modify Configuration

All vFabric 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/vmware/vfabric-web-server/myserver`.

For example, the `httpd-info.conf` sample configuration file shows how you can get information about the requests being processed by the vFabric 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 vFabric 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.

Configure Load Balancing Between Two or More tc Runtime Instances

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

In the procedure that follows, you configure a vFabric Web Server instance to run *in front* of the tc Runtime instances; this vFabric 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 vFabric Web Server instance. For this reason, the vFabric 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 vFabric 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:
  - `/home/tcserver/vfabric-tc-server-standard/instanceOne`
  - `/home/tcserver/vfabric-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 vFabric 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.
- vFabric Web Server is installed on a different computer than vFabric tc Server. The name of the particular vFabric Web Server instance is `lb-server` and its home directory is `/opt/vmware/vfabric-web-server/lb-server`.
- You want to configure *sticky sessions*, which means that the vFabric 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 vFabric 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 vFabric Web Server instance and the two tc Runtime instances.

Prerequisites

- Install vFabric Web Server on your platform and create a new instance.
- Install vFabric tc Server on the same or different computer as vFabric Web Server, and create two more instances. Make note of the host and port numbers of the two instances. See the vFabric 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 vFabric Web Server is installed, stop the instance, if it is currently running. Following the example and assumptions:

   ```
   prompt# cd /opt/vmware/vfabric-web-server/lb-server
   prompt# bin/httpdctl stop
   ```

2. Open the `httpd.conf` configuration file of the vFabric 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/modules/mod_proxy_balancer.so"
   ```

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

where *VFWS-INSTALL* refers to the directory in which you installed vFabric 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/vmware/vfabric-web-server/httpd-2.2/modules/mod_proxy_balancer.so"
LoadModule proxy_module           "/opt/vmware/vfabric-web-server/httpd-2.2/modules/mod_proxy.so"
LoadModule proxy_http_module      "/opt/vmware/vfabric-web-server/httpd-2.2/modules/mod_proxy_http.so"
```

The vFabric Web Server configuration file is located in the `conf` directory of your vFabric Web Server instance (`/opt/vmware/vfabric-web-server/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 vFabric 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 vFabric 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 vFabric 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 vFabric 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 vFabric 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 vFabric 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, VMware 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 vFabric tc Server documentation.

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

```
prompt# cd /opt/vmware/vfabric-web-server/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 /home/tcserver/vfabric-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 vFabric 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 vFabric 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 vFabric Web Server instance rather than the tc Runtime instance. For example, if the URL to access the vFabric 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 vFabric Web Server instance to route requests to the `/my-app` application to one of the two tc Runtime instances. The vFabric 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 vFabric 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 vFabric Web Server instance is actually running.

6. Additional Documentation

The documentation in this guide provides information about what vFabric Web Server contains; how to install it; and how to create, start, and stop instances. The vFabric 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 http://httpd.apache.org/docs/2.2/
- OpenSSL http://www.openssl.org/docs/
- PERL http://perldoc.perl.org/
- Mod_perl http://perl.apache.org/
- PHP http://www.php.net/docs.php
- Mod_SSL (for Apache HTTP Server 1.3) http://www.modssl.org/docs/2.8/
- 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